Assessment of the One-Shot Strategy for the Calibration of Marine Ecosystem Models

Dissertation

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Dr. rer. nat.)

der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von
Dipl.-Math. Claudia Kratzenstein

Kiel 2015

Erstgutachter: .... Prof. Dr. Thomas Slawig

Zweitgutachterin: .... Prof. Dr. Andrea Walther

Tag der mündlichen Prüfung: 17.7.2015
Zum Druck genehmigt: 17.7.2015

gez. Prof. Dr. Eckhard Quandt, Dekan

Zusammenfassung

Das Optimieren von Modellparametern ist eine äußerst wichtige und hochaktuelle Aufgabe in der Entwicklung von Klimamodellen oder Modellen, die Teile des Klimasystems simulieren. Diese Modelle sollen verlässliche Aussagen über das zukünftige Klima liefern. Dafür werden berechnete Modellvariablen an vorhandene Messdaten oder an Ausgaben anderer Modelle ausgerichtet, indem Modellparameter geeignet korrigiert werden. Dies entspricht einem mathematischen Minimierungsproblem mit Nebenbedingungen in Form von partiellen oder gewöhnlichen Differentialgleichungen. Häufig liegt der Lösung der Modellgleichungen ein iterativer Prozess zugrunde, da analytische Lösungen oft nicht gegeben sind. Konventionelle Optimierungsalgorithmen führen für jede Funktionsauswertung diesen iterativen Prozess (oder Fixpunktlöser) durch, der allein schon erheblichen Rechenaufwand und Rechenzeit von Stunden bis zu Tagen benötigen kann. Im gesamten Optimierungsprozess werden je nach Methode zahlreiche Funktionsauswertungen und gegebenenfalls Ableitungsinformationen benötigt, um Parameter geeignet zu adjustieren.

In dieser Arbeit wird die One-shot Optimierungsmethode nach Hamdi und Griewank erstmals für die Kalibrierung zweier illustrativer mariner Ökosystemmodelle untersucht. Die One-shot Methode korrigiert Parameter bereits in jedem Schritt des Fixpunktlösers, der für die Berechnung der Lösung der Modellgleichungen benutzt wird. Dabei wird der Fixpunktlöser durch eine Aufdatierung des adjungierten Zustandes und der Parameterkorrektur erweitert und soll schließlich während einer Fixpunktiteration (mit nur wenig Verzögerung) eine zulässige Lösung der Modellgleichungen als auch optimale Parameter liefern.

In der vorliegenden Arbeit wird geprüft, ob die One-shot Methode für Modelle des marinen Ökosystems anwendbar ist in Bezug auf theoretische Voraussetzungen der Konvergenztheorie, Implementierbarkeit und ihrer Güte bezüglich der berechneten Lösungen und der Effizienz des Algorithmus'. Großen Wert hat die Beschreibung der Anwendung auf ein Modell mit instationärer (hier jährlich periodischer) Lösung. Es wird eine Anleitung zur Implementierung erstellt und mögliche Vereinfachungen und geeignete Einstellmöglichkeiten empfohlen.

Anhand der vorgestellten Ergebnisse können der Aufwand, die Vorteile und die Nachteile der One-shot Methode in Zusammenhang mit marinen Ökosystemmodellen klar aufgezeigt werden.

Abstract

Parameter optimization is an important and highly topical task in all kinds of climate models or models that simulate parts of the climate system. These models are to provide reliable projections on the future climate. Computed model variables are fit to measurements or to output from other models by correcting model parameters. This is a mathematical minimization problem with constraints in the form of nonlinear ordinary or partial differential equations. Since analytical solutions often cannot be provided, usually an iterative method is applied to solve the model equations. Classical optimization strategies perform this iterative method (or fixed point solver) for each function evaluation. The fixed point solver may need extensive computing time lasting hours or even days. Depending on the optimization approach, each optimization run requires numerous function evaluations and if necessary derivative information to adjust parameters.

In this work, for the first time the application of the One-shot optimization strategy according to Hamdi and Griewank is investigated for the calibration of two illustrative marine ecosystem models. The One-shot method corrects parameters in each step of the iterative process applied for solving the model equations. The fixed point iteration is augmented by an update of the adjoint state and the correction of the parameters. It aims at computing a feasible state and optimal parameters with only bounded retardation compared to the underlying fixed point iteration with fixed parameters.

This work examines the applicability of the One-shot strategy in the calibration of marine ecosystem models with respect to the theoretical assumptions of the convergence theory, its implementation, the quality of computed results and the efficiency of the algorithm. The application to a model with unsteady (here annually periodic) PDEs is of great value. Instructions for the implementation, simplifications and suitable adjustments are presented.

The numerical results identify the costs, the advantages and the disadvantages of the One-shot optimization technique in the calibration of marine ecosystem models.

Contents

1 Introduction

Mathematical modeling has become indispensable in a growing number of scientific fields. Mathematical models are employed to explain and simulate real life phenomena or to predict the effects on these under certain changes of conditions.
Among those scientific fields is the wide topic of climate modeling. It deals with the modeling of processes in the different components of the earth's system, as for example the atmosphere, the oceans and other water masses, land and sea ice, and terrestrial and marine ecosystems to name only some of them (see for example McGuffy et al. [24]). It considers processes on temporal and spacial scales which reach from small (e.g. cloud building) to very large (e.g. ocean currents) scales.
The subject of climate modeling is closely related to the topic of global climate change which has become one of the most pressing challenges our society is facing at the moment. It is a topic which conglomerates politicians, scientists, economists and of course individuals and their lifestyles.
It is essential to estimate future changes in temperatures, the sea level, atmospheric components, changes in nutrient supply and associated biological productivity as well as changes in the oceanic carbon uptake and associated acidification to name only some of the issues. There is the inevitable need for mathematical models on a high resolution for regional predictions.

Still, some processes are not well-known, some are too small-scaled in time or space, and others are just beyond the scope of the model. All these processes are *parameterized*, i.e. simplified model functions (parameterizations) are used. These necessarily include lots of – most of the time – only heuristically known parameters. A main task thus is to *calibrate* the models by optimizing the parameter w.r.t. data from measurements or other (more complex) models.

Similar to many applications in engineering applications of fluid mechanics, also in geophysical flows (e.g. ocean models) an optimization is at first performed for steady states of the equations before proceeding to transient problems. This means that only the stationary solution is used in the cost or objective function to be minimized. Moreover (and this is the second point where engineering and geophysical flow problems are similar), the computation of steady states is often performed by running a transient model into the steady state. This strategy is called pseudo time stepping, since the time variable may be regarded as a kind of iteration counter.

It is well known from optimal control of differential equations that the classical adjoint technique (that allows the representation of the gradient of the cost) leads to a huge amount of recomputations, storing or both. This problem looks even more frustrating in the pseudo-time stepping context, since here only the final, numerically converged state is important for the cost. Nevertheless a classical adjoint technique would need all intermediate iterates.

If the number of parameters to be optimized is small, a sensitivity equation approach is also reasonable. On the discrete level this is comparable to the application of the forward mode of Automatic or Algorithmic Differentiation (AD). Here, the sensitivity equation has the same temporal integration direction (namely forward) as the original pseudo time stepping. But nevertheless it is worthwhile investigating how the two (for a non-linear model) coupled iterations for state and sensitivity are performed.

Griewank and Walther describe in [14], chapter 15, the differences between two-phase (where the iteration for the state is run to the steady state or fixed point first, and then the sensitivity is computed) and piggy-back approaches (where both iterations are combined to one). Christianson in [4] proposed to perform the sensitivity iteration with the converged state instead of using its iterates. Giering, Kaminski and Vossbeck in [18] used the so-called *Full Jacobian approach*, where they directly used the steady state equation and differentiated it to obtain an equation for the gradient.

The approach used here is called (Jacobi) One-shot approach, which was in this form developed by Griewank and Hamdi [10, 11], and can be seen as an extension of the piggy-back strategy aiming for optimality and feasibility simultaneously with the so-called bounded retardation. That means that the number of One-shot iteration steps shall not too much exceed the number of fixed point iteration steps that are necessary for the computation of a feasible state. Theoretical results were published in [10, 11] and [7, 2], an engineering application was presented by Özkaya and Gauger in [26]. Among further developments of this original Jacobi One-shot approach is the Multi-step (Seidel) One-shot method [2] where after one design update several repeated state updates are followed by the same number of repeated adjoint updates.

The idea of simultaneous solution of state equations and parameter correction is not new. In [38], Ta'asn uses a pseudo-time embedding for the state and adjoint state equations and the design equation is solved as an additional boundary condition. This still results in a differential algebraic equation which requires some strategy to solve the design equation alone.

In [16], the authors construct a system of only ODEs which is solved by a time stepping method in the spirit of reduced SQP-methods. They develop a preconditioner working on the whole system of equations with state, adjoint state and design equations.

In the One-shot approach used here, the idea is that for fixed parameters there is a given (not necessarily (pseudo-) time stepping) strategy to solve the state equations. This strategy is assumed to require no or disallow any changes. In each iteration step the update of the state is augmented by an update of the adjoint state and a kind of quasi-Newton step for the design correction with the distinctive feature that the required preconditioner controls convergence of the whole system. Here, the preconditioner is a squared matrix of only the size of the number of parameters.

Since the assumptions in the theoretical analysis of the One-shot method are very strict and the computation of the preconditioner seems at first glance laborious and expensive, the intention of this work is to investigate the applicability of the One-shot strategy for marine ecosystem models and possibly propose simplifications. We analyze the method applied to a low-complexity box model of the North Atlantic and a model of medium complexity simulating the phosphorus cycle in the world's oceans. For the low-complexity model, we compare numerical results to the gradient based BFGS and limited-memory BFGS (L-BFGS) methods ([25], pp. 136-143 and 176 - 181). We set aside the comparison to genetic or so-called intelligent search algorithms, see e.g. [29], because the aim of the One-shot approach according to the authors of [10, 11] is to offer an alternative to local gradient-based optimization techniques. Genetic algorithms usually require a high number of function evaluations which we want to avoid because of the costly computation of feasible states needed for the function evaluation.
For the phosphorus cycle model (N-DOP model) we convert computing time into the number of full computations of feasible states by the pseudo time stepping strategy as an indicator of quality in the comparison to methods requiring a feasible state for each function evaluation.

The problems considered here are different from the application in [26] in that the parameters enter in a nonlinear fashion resulting in so-called non-separable adjoints where the adjoint is no longer only the sum of a term on the state and a term on design. Furthermore, as the N-DOP model simulates the annual cycle of phosphate and dissolved organic phosphorus, the search for the feasible unsteady state here is the search for a steady annual cycle considering the periodicity of the model. The approach presented here differs from the one described in Günther et al. [15, 2] in that we do not augment the contractive fixed-point iterator but augment the cost functional in an appropriate way.

The outline of this thesis is the following. The subsequent chapter 2 summarizes the derivation of the (original) One-shot strategy according to Griewank and Hamdi [10, 11]. In addition, we provide a very short introduction to Algorithmic or Automatic Differentiation (AD) as our implementation of the One-shot strategy relies on the application of AD.
In chapter 3, we embrace the results of the investigations concerning the calibration of the box model of the North Atlantic. The integration scheme to compute a steady state is a full step explicit Euler time stepping. One Euler step corresponds to one model year.
In the main part of this work, chapter 4, we analyse the One-shot approach in the calibration of the spatially three-dimensional N-DOP model. Its time integration scheme with 2,880 intermediate time steps per model year poses new challenges with respect to the applicability and efficiency of the One-shot method.
The final chapter 5 summarizes results and we define open tasks for future investigations.

2 The One-shot Optimization Strategy

In this chapter, we recapitulate the One-shot optimization strategy according to Griewank and Hamdi [10, 11], its quintessence and difference to conventional optimization methods, and we derive and explain the One-shot iteration step.

2.1 Problem Formulation

Parameters $u \in U$ of a model describing physical, biological, chemical or other real life phenomena are usually determined by fitting model output $y = y(u) \in Y$ to observed data denoted by y_{data}. This data can also be taken from other, more comprehensive models.

The fitting procedure then is a mathematical optimization problem with a least-squares cost functional with some regularization term

$$J(y,u) = \frac{1}{2}\|y - y_{data}\|_Y^2 + \frac{\alpha}{2}\|u - u_{est}\|_U^2, \qquad \alpha \in \mathbb{R}_0^+$$

under the constraint that model equations are fulfilled. We write:

$$\min_{y,u} J(y,u) \text{ s.t. } c(y,u) = 0. \tag{2.1}$$

In climate modeling, model equations are usually partial and/or ordinary differential equations solved by an iterative process.

In this work, we will use the equivalent denominations *objective* or *objective function* for the cost functional J, *design* or *control vector* for the the parameter set u and *state (vector)* for the model output y.

The problem (2.1) will become more difficult with respect to uniqueness of minima and computation of derivative information, if the quantity to be fit is computed from a functional $g : (Y,U) \to Y_1$ such that J then is

$$J(y,u) = \frac{1}{2}\|g(y,u) - g_{data}\|_{Y_1}^2 + \frac{\alpha}{2}\|u - u_{est}\|_U^2$$

or more general with a function $F : (Y, U) \to Y_2$

$$J(y,u) = \frac{1}{2}\|F(y,u)\|^2_{Y_2} + \frac{\alpha}{2}\|u - u_{est}\|^2_U$$

With the help of the regularization term $\frac{\alpha}{2}\|u - u_{est}\|^2_U$ parameters u are kept in an acceptable or estimated range around parameter values u_{est}, where elements $u_{est,i}$ can for example be taken as mean values of some maximum and minimum values.
We assume J to be $C^{2,1}$, i.e. twice continuously differentiable in y and once in u.

2.2 One-shot Iteration and its Properties

In practice, analytically finding a feasible state y^* with $c(y^*, u) = 0$ often is impossible. Therefore, usually an iterative method $G : Y \times U \to Y$ is called upon with

$$G(y_k, u) \overset{k\to\infty}{\longrightarrow} y^* = y^*(u). \tag{2.2}$$

For the One-shot optimization approach, we assume that there is a given fixed point iteration, also called model spin-up, which has already been found reliable and successful in the search for the feasible state y^* for parameters u. Included step size or preconditioner strategies can be carried over and do not have any influence on the One-shot iteration. Thus, there is a *given* contraction, (pseudo-) time stepping strategy or fixed point iteration G, for which y^* satisfies $y^* = G(y^*, u) = \lim_{k\to\infty} G(y_k, u)$.
In [10, 11], the authors assume the iteration function $G : Y \times U \to Y$ being $C^{2,1}$ with the contraction factor $\rho < 1$, i.e. for a suitable inner product norm $\|\cdot\|$ we have for G_y, denoting the Jacobian of G with respect to y, that

$$\|G_y(y,u)\| \le \rho < 1, \qquad \forall y \in Y. \tag{2.3}$$

With the mean value theorem (see e.g. Nocedal and Wright [25], pp. 629–630) follows

$$\|G(y_1,u) - G(y_2,u)\|_Y \le \rho\|y_1 - y_2\|_Y, \qquad \forall y_1, y_2 \in Y. \tag{2.4}$$

With this contraction property of G we can infer from the Banach fixed point theorem (Papageorgiou et al. [27], p. 226), for fixed u, the sequence $y_{k+1} = G(y_k, u)$, $k \to \infty$, converges to a unique limit $y^* = y^*(u)$.
With the help of the fixed point reformulation, the optimization problem (2.1) can be

written as

$$\min_{y,u} J(y,u) \quad s.t. \quad y = G(y,u). \tag{2.5}$$

A conventional optimization strategy performs a complete model spin-up with parameters u_k in each iteration step $k \geq 0$ obtaining the feasible state $y^*(u_k) = \lim_{l\to\infty} G(y_l, u_k)$, computes $J(y^*(u_k), u_k)$ and adjusts model parameters obtaining u_{k+1}.
Adjusting the parameters demands further full model spin-ups and/or expensive derivative information. Computing the gradient $\frac{dJ}{du}(y^*(u_k), u_k)$ is challenging as it has to be computed iteratively as well. For instance, the finite differences (FD) approach needs to search for additional feasible states $y^*(u_k + he_i)$, where e_i, $i = 1, ..., \dim(u)$, are the unit vectors and h is the step size determining the accuracy of the FD approach for dJ/du. As $y^*(u_k + he_i)$ are in turn approximations with respect to a certain accuracy of the fixed point iteration and its termination condition, estimating the truncation error of the gradient $\frac{dJ}{du}(y^*(u_k), u_k)$ is difficult. Carrying forward exact derivatives in hand with the model spin-up (so-called piggyback approach described in Griewank and Walther [14], chapter 15) at least eliminates the inaccuracy of the FD approach. However, the question remains if the gradient converges as fast as the state. In [14] a *time-lag* is observed. Furthermore, these exact derivatives need to be provided.

In high-complexity long term simulation models, one spin-up can take up to weeks of computational time depending on the computational platform. Therefore, the main goal of the One-shot strategy is to adequately adjust model parameters already during the fixed-point iteration process. Its update rules are deduced from the Lagrange multipliers approach.
In the following, we assume the considered space to have finite dimensions corresponding to their discretization, s.t. $Y \subset \mathbb{R}^n$ and $U \subset \mathbb{R}^m$. This allows us to write inner products as scalar products in Euclidean space and adjoint states as transposed vectors. The associated Lagrangian to problem (2.5) with the Lagrange multiplier or adjoint state $\bar{y} \in \bar{Y} \subset \mathbb{R}^n$ is

$$L(y, \bar{y}, u) = J(y,u) + \bar{y}^\top (G(y,u) - y).$$

A Karush-Kuhn-Tucker (KKT) point $(y^*, \bar{y}^*, u^*)$ fulfilling the first order necessary optimality condition ([25] p. 321) must satisfy

$$\left.\begin{aligned} 0 = \tfrac{\partial L}{\partial y} = L_y &= J_y(y^*, u^*) + \bar{y}^{*\top} G_y(y^*, u^*) - \bar{y}^{*\top}, \\ 0 = \tfrac{\partial L}{\partial \bar{y}} = L_{\bar{y}} &= G(y^*, u^*) - y^*, \\ 0 = \tfrac{\partial L}{\partial u} = L_u &= J_u(y^*, u^*) + \bar{y}^{*\top} G_u(y^*, u^*). \end{aligned}\right\} \quad (2.6)$$

Motivated by this system of equations, the following coupled full step iteration, called *One-shot strategy*, to reach a KKT point is derived:

```
Initialize $\mathtt{y_0}$, $\mathtt{\bar{y}_0}$ and $\mathtt{u_0}$, k=0, define $\varepsilon_{\mathtt{tol}}$
do until $\|\mathtt{L}_{(\mathtt{y},\bar{\mathtt{y}},\mathtt{u})}\| < \varepsilon_{\mathtt{tol}}$ (or an alternative termination condition is fulfilled):

    $\mathtt{y_{k+1}} = \mathtt{G(y_k, u_k)}$                                                  // towards feasibility
    $\mathtt{\bar{y}_{k+1}} = \mathtt{J_y(y_k, u_k) + \bar{y}_k^\top G_y(y_k, u_k)}$                // towards adjoint feasibility     (2.7)
    $\mathtt{u_{k+1}} = \mathtt{u_k - B_k^{-1}(J_u(y_k, u_k) + \bar{y}_k^\top G_u(y_k, u_k))}$   // towards optimality
    $\mathtt{k} = \mathtt{k+1}$
```

ε_{tol} is the tolerance at which a solution is accepted. Alternatively, other stopping criteria can be set, for example by a fixed iteration counter or if there is numerically no change in the updated variables anymore.

Note that here derivatives of J and G depend on the current iterates y_k, u_k only, and y_k and u_k are considered independent variables of J and G. That means in the One-shot context $\frac{\partial y_k}{\partial u}(u_k) = 0$ whereas (in general) $\frac{\partial y_{k+1}}{\partial u}(u_k) \neq 0$ as we equivalently denote $G_u(y_k, u_k) = \frac{\partial G}{\partial u}(y_k, u_k) = \frac{\partial y_{k+1}}{\partial u}(u_k)$. We carefully distinguish between $J_u(y^*(u_k), u_k)$ in the context of conventional optimization strategies and $J_u(y_k, u_k)$ in the One-shot context. We emphasize that in contrast to conventional optimization strategies, no full model spin-ups with intermediate parameter sets u_k are needed in the One-shot method.

In the update of u_k, the matrix $B_k \in \mathbb{R}^{m \times m}$ is a design space preconditioner which must be selected to be symmetric positive definite. As mentioned above, we do not want to introduce an additional preconditioner for the update of the state y, because of the assumption that the model spin-up G has already been found reliable and successful in the search for feasible states. Possible preconditioners or step size corrections are assumed to be included in the update function G.

The contractivity (2.4) ensures that the first equation in the coupled iteration step

(2.7) converges ρ-linearly for fixed u. Although the second equation exhibits a certain time-lag, it converges with the same asymptotic R-factor [13]. Therefore, we neither introduce a preconditioner for the update of $\bar{y}$. As far as the convergence of the coupled iteration (2.7) is concerned, the goal is to find B_k that ensures that the spectral radius of the coupled iteration (2.7) stays below 1 and as close as possible to ρ. In subsection 2.3, we recall the formula of appropriate preconditioners B_k according to Griewank and Hamdi [10, 11].

In the context of the One-shot optimization, we distinguish between problems with so-called separable and and those with non-separable adjoints. The mixed derivative $L_{yu} = J_{yu} + \bar{y}^\top G_{yu}$ represents the sensitivity of the adjoint equation with respect to design parameters. A problem where $L_{yu} = 0$ is separable and where $L_{yu} \neq 0$ is called non-separable. In [10, 11] the authors concentrate on separable problems. In climate modeling, the model equations usually include a non-linear coupling of the state and design variables, such that generally $L_{yu} \neq 0$ holds. The problems we consider in chapters 3 and 4.5 not only have non-linear couplings in the model equations but also in the cost functionals.

2.3 Preconditioner B and the doubly augmented Lagrangian

In this section, we explain the choice of the preconditioners B_k according to [10, 11]. For the sake of simplicity, we omit the iteration index k using the notations B, y, $\bar{y}$ and u only.

For the derivation of the preconditioner B, we introduce the doubly augmented Lagrangian L^a

$$L^a(y, \bar{y}, u) = L(y, \bar{y}, u) + \frac{\alpha_L}{2} \|G(y, u) - y\|_Y^2 + \frac{\beta_L}{2} \|L_y(y, \bar{y}, u)\|_Y^2,$$

which is the Lagrangian of the original problem augmented by the errors in primal and dual feasibility. $\alpha_L > 0$ and $\beta_L > 0$ are weighting coefficients.

The authors of [10] prove that under certain conditions on α_L and β_L (see below (2.9) and (2.11)), stationary points of problem (2.5) are also stationary points of L^a and that L^a is an exact penalty function. This leads to the idea to choose B as an approximation to the Hessian of L^a, i.e. $B \approx \nabla_{uu} L^a$.

In [10], it is proven that descent of the augmented Lagrangian is provided for any preconditioner B fulfilling

$$B \succeq B_0 \text{ for } B_0 := \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L L_{yu}^\top L_{yu}) \tag{2.8}$$

and where $B \succeq B_0$ denotes $B - B_0$ being positive semidefinite. σ is defined as

$$\sigma := 1 - \rho - \frac{(1 + \frac{\|L_{yy}\|}{2}\beta_L)^2}{\alpha_L \beta_L (1-\rho)} \tag{2.9}$$

to ensure convergence of the coupled iteration and where α_L and β_L need to be selected appropriately.

The authors of [10] propose to choose α_L and β_L such that B_0^{-1} is as large as possible for large parameter corrections. Using (2.8) we get

$$\|B_0\|_2 \leq \frac{1}{\sigma}(\alpha_L \|G_u\|_2^2 + \beta_L \|L_{yu}\|_2^2).$$

Minimizing the right most formula as a function of α_L and β_L and replacing σ with (2.9) yields proposition 3.3 from [10] which concludes:

Under the assumption that

$$\sqrt{\alpha_L \beta_L}(1-\rho) > 1 + \frac{\beta_L}{2}\|L_{yy}\| \tag{2.10}$$

holds and $\|L_{yy}\| \neq 0$, for B_0 from (2.8) choose

$$\beta_L = \frac{3}{\sqrt{\|L_{yy}\|^2 + 3\frac{\|L_{yu}\|^2}{\|G_u\|^2}(1-\rho)^2} + \frac{\|L_{yy}\|}{2}} \text{ and } \alpha_L = \frac{\|L_{yu}\|^2}{\|G_u\|^2} \frac{\beta_L(1 + \frac{\|L_{yy}\|}{2}\beta_L)}{(1 - \frac{\|L_{yy}\|}{2}\beta_L)}. \tag{2.11}$$

In the separable case where $\|L_{yu}\| = 0$, it is $\beta_L = 2/\|L_{yy}\|$. Inserting the formula of β_L from (2.11) into α_L from (2.11) and afterwards setting $\frac{\|L_{yu}\|^2}{\|G_u\|^2} = 0$ for the separable case, yields

$$\beta_L = \frac{2}{\|L_{yy}\|}, \quad \alpha_L = \frac{4\|L_{yy}\|}{(1-\rho)^2} \text{ and } \sigma = \frac{(1-\rho)}{2}. \tag{2.12}$$

Striving for approximating B as $B \approx \nabla_{uu} L^a$, we constitute that at a stationary point of L^a, where primal and dual feasibility hold, the Hessian of L^a, namely $\nabla_{uu} L^a$, is

$$\nabla_{uu} L^a = \alpha_L G_u^\top G_u + \beta_L L_{yu}^\top L_{yu} + L_{uu}.$$

As L^a is an exact penalty function, we have $\nabla_{uu} L^a \succ 0$ in a neighbourhood of the constrained optimization solution. Assuming that $L_{uu} \succ 0$ implies that the preconditioner

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L L_{yu}^\top L_{yu} + L_{uu}) \tag{2.13}$$

fulfills (2.8) and thus the step $s_k := u_{k+1} - u_k = -B^{-1} L_u(y_k, \bar{y}_k, u_k)$ of the coupled iteration (2.7) yields descent on L^a.

In the more recent paper [11], the authors perform a different approach in the choice of the weighting factors. They set σ to $\sigma = 1$, compute the weights as

$$\alpha_L = \frac{2\|L_{yy}\|}{(1-\rho)^2}, \qquad \beta_L = \frac{2}{\|L_{yy}\|} \tag{2.14}$$

or in a third version as

$$\alpha_L = \frac{6\|L_{yy}\|}{(1-\rho)^2}, \qquad \beta_L = \frac{6}{\|L_{yy}\|}, \tag{2.15}$$

but in this case need to perform a (standard backtracking) line search procedure to enforce convergence.

To even more simplify the computations of the weights, the authors propose to fix $\|L_{yy}\|$ to 1, since iterative strategies as the power iteration (compare Golub and Van Loan [9], section 8.2) always converged to 1.

In our testings with climate model examples, we constitute that the choice of the weights has a significant influence on the performance of the One-shot optimization.

We have emphasized the One-shot method not needing to perform full spin-ups for intermediate parameters u_k. Still a lot of first and second order derivatives of J and G are needed. As we strive for exact derivatives, we apply automatic differentiation tools.

2.4 Automatic Differentiation

Automatic Differentiation (AD) is a software technology to compute the derivative of a function at costs of only a small multiple of the costs for the evaluation of the function itself. With the help of source code transformation or operator overloading an AD tool provides the user with a computer programme containing the derivatives.

Automatic differentiation tools are for example TAF or ADiMat, which use the source code transformation approach to generate FORTRAN or MATLAB subroutines to calculate function values and derivative information in one call, see [8, 17], or for example

ADOL-C using the operator overloading concept in C/C++ codes, see [12].
Both concepts split up the function evaluation code into elementary functions and intermediate values. They internally draw up an *evaluation trace*. For elementary functions the (directional) derivative is known such that with the help of the chain rule, the derivative of the implemented function can be calculated.

There are two modes of AD: the forward and the reverse mode. The forward mode simultaneously evaluates and carries forward a directional derivative of each intermediate variable. Then for some direction $z \in \mathbb{R}^N$, a function $f : \mathbb{R}^N \to \mathbb{R}^M$ and the input variable $x \in \mathbb{R}^N$ the forward mode yields the vector

$$v = f'(x)z \in \mathbb{R}^M$$

where $f'(x)$ is the Jacobian of f. Here, the costs for evaluating derivatives with the forward mode increase linearly with the number of domain directions z along which the user wants to differentiate. Hence, one benefits from a small number of independent variables.

With the reverse mode of AD one can also benefit from a small number of dependent variables.
The reverse mode stores all intermediate variables of the evaluation trace. Then, in a reverse sweep partial derivatives of the function with respect to intermediate variables are accumulated. For a vector $v \in \mathbb{R}^M$, the reverse mode computes the vector

$$z^T = v^T f'(x) \in \mathbb{R}^N. \tag{2.16}$$

In particular, the gradient of a scalar valued function $f : \mathbb{R}^N \to \mathbb{R}$ can be obtained in only one reverse sweep.
The concatenation of a reverse and a forward sweep yields second order derivatives.

For the One-shot optimization strategy generally applying AD is not recommended. $\bar{y}^\top G_y$ and $\bar{y}^\top G_u$ seem to be suitable for the reverse mode of AD. Still, depending on the considered model and its iterative solver, the application of automatic differentiation is not as *automatic* as the name suggests. We are going to describe the utilization of AD in the two considered models, the 4-box model of the North Atlantic and the N-DOP model, in chapters 3 and 4 respectively.

3 Calibration of a 4-Box-Model of the North Atlantic

To investigate the One-shot strategy with respect to the applicability on optimization problems with models used in marine sciences we chose the Rahmstorf 4-box-model of the thermohaline circulation of the North Atlantic [40] for first testings. Besides fast computation of feasible and/or optimal states, the advantage of using this low-complexity model is the possibility to easily apply standard optimization strategies which are generally provided by optimization toolboxes for certain programming languages.

3.1 Model Description

The considered box-model simulates the flow rate or the *meridional volume transport* of the Atlantic Ocean known as the *conveyor belt*, carrying heat northward and having a significant impact on climate in northwestern Europe. Temperatures $T_i = T_i(t)$ and salinity differences $S_i = S_i(t)$ in four different boxes $i = 1, ..., 4$, the southern, northern, tropical and the deep Atlantic, are the characteristics inducing the flow rate m. The surface boxes exchange heat and freshwater with the overlying atmosphere, which causes a pressure-driven circulation. The time dependent ordinary differential equation (ODE) system reads

$$\left.\begin{array}{lrlr}
\frac{dT_1}{dt} = \dot{T}_1 = \lambda_1(T_1^* - T_1) & + \frac{m}{V_1}(T_4 - T_1) & \frac{dS_1}{dt} = \dot{S}_1 = \quad \frac{S_0 f_1}{V_1} & + \frac{m}{V_1}(S_4 - S_1) \\
\frac{dT_2}{dt} = \dot{T}_2 = \lambda_2(T_2^* - T_2) & + \frac{m}{V_2}(T_3 - T_2) & \frac{dS_2}{dt} = \dot{S}_2 = -\frac{S_0 f_2}{V_2} & + \frac{m}{V_2}(S_3 - S_2) \\
\frac{dT_3}{dt} = \dot{T}_3 = \lambda_3(T_3^* - T_3) & + \frac{m}{V_3}(T_1 - T_3) & \frac{dS_3}{dt} = \dot{S}_3 = \frac{S_0(f_2 - f_1)}{V_3} & + \frac{m}{V_3}(S_1 - S_3) \\
\frac{dT_4}{dt} = \dot{T}_4 = & \frac{m}{V_4}(T_2 - T_4) & \frac{dS_4}{dt} = \dot{S}_4 = & \frac{m}{V_4}(S_2 - S_4)
\end{array}\right\}(3.1)$$

$$m = k_m(\beta_m(S_2 - S_1) - \alpha_m(T_2 - T_1))$$

where m is the meridional volume transport or overturning.

The constants f_1, f_2 are freshwater fluxes containing atmospheric water vapor trans-

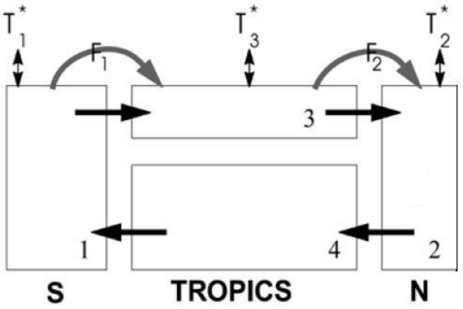

Figure 3.1: Illustration of the Rahmstorf 4-box model described in Zickfeld et al. [40]. Shown is the flow direction with an upwelling in the south and downwelling in the north, i.e. $m > 0$.

port and wind-driven oceanic transport. They are multiplied by a reference salinity S_0 for conversion to a salt flux. Moreover, k_m is a coupling constant of flow, α_m, β_m are expansion coefficients and T_i^* temperatures towards which the surface boxes $i = 1, ..., 3$ are relaxed. λ_i are thermal coupling constants and V_i volumes of boxes $i = 1, ..., 4$. Figure 3.1 illustrates the flow.

An explicit full step Euler time stepping is chosen to calculate steady states for the temperatures T_i and salinity differences S_i from which the overturning m is computed. If the state vector $y = (T_1, T_2, T_3, T_4, S_1, S_2, S_3, S_4)$ gathers the temperatures and salinities, the spin-up function $G : Y \times U \to Y$, $Y \subset \mathbb{R}^8$, $U \subset \mathbb{R}^{\dim(u)}$ reads

$$G(y, u) = y + F_{Box}(y, u),$$

where $F_{Box} : Y \times U \to Y$ denotes the right-hand side of the ODE system 3.1 and u is the considered parameter vector, which contains tunable model parameters which will be defined in the course of this section.

In this version from Zickfeld et al. [40] only positive m are admissible, because otherwise the model equations are wrong. If the computed value of m is negative during the Euler spin-up, the authors set m to zero. As the One-shot strategy assumes the spin-up function G to be $C^{2,1}$ this version of the box-model is not applicable.

In [39] a smooth coupling of the two possible flow directions is proposed. The formulation of the time dependent ODE system there is

$$\left.\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m^+}{V_1}(T_4 - T_1) + \tfrac{m^-}{V_1}(T_3 - T_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m^+}{V_2}(T_3 - T_2) + \tfrac{m^-}{V_2}(T_4 - T_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m^+}{V_3}(T_1 - T_3) + \tfrac{m^-}{V_3}(T_2 - T_4)\\
\dot{T}_4 &= \tfrac{m^+}{V_4}(T_2 - T_4) + \tfrac{m^-}{V_4}(T_1 - T_4)\\
\dot{S}_1 &= \tfrac{S_0 f_1}{V_1} + \tfrac{m^+}{V_1}(S_4 - S_1) + \tfrac{m^-}{V_1}(S_3 - S_1)\\
\dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m^+}{V_2}(S_3 - S_2) + \tfrac{m^-}{V_2}(S_4 - S_2)\\
\dot{S}_3 &= \tfrac{S_0(f_2 - f_1)}{V_3} + \tfrac{m^+}{V_3}(S_1 - S_3) + \tfrac{m^-}{V_3}(S_2 - S_4)\\
\dot{S}_4 &= \tfrac{m^+}{V_4}(S_2 - S_4) + \tfrac{m^-}{V_4}(S_1 - S_4)
\end{aligned}\right\} \tag{3.2}$$

$$m = k_m(\beta_m(S_2 - S_1) - \alpha_m(T_2 - T_1))$$

where for some positive a, $m^+ = \frac{m}{1-e^{-am}}$ almost coincides with the meridional volume transport or overturning m for positive m and is almost zero for negative m. The term $m^- = \frac{-m}{1-e^{am}}$ becomes almost zero for positive m and $-m$ for negative m. That means the summands including m^+ and m^- are activated or deactivated depending on the flow direction. The deviation from the physically correct model becomes smaller the larger a is.

With this modification, the system is smoothed but still we have to exclude pairs (T_1, T_2, S_1, S_2) where m becomes zero. In our testings with the Euler spin-up, for reasonable starting values, we never computed temperatures and salinities where $T_1 = T_2$ and $S_1 = S_2$, or where $0 = \beta_m(S_2 - S_1) - \alpha_m(T_2 - T_1)$.

Several model parameters are involved, the most important being the freshwater flux f_1 containing atmospheric water vapor transport and wind-driven oceanic transport; it is used to simulate global warming in the model and is chosen in the interval $[-0.2, 0.15]$. T_i^*, $i = 1, 2, 3$, are so-called restoring temperatures, which can be seen as counterparts of the three surface temperatures. Further model parameters are physical, relaxation and coupling constants among which there are well-known fixed parameters and those which are tunable parameters. See [40] for an explanation of the occurring constants, fixed parameters and tunable parameters.

3.2 Fitting Data from the Climber2 Model

Given fresh water fluxes $(f_{1,i})_{i=1}^{l}$, corresponding to different warming scenarios, the aim is to fit the overturning values $m_i = m(y(f_{1,i}), u)$ computed from stationary temperatures and salinities $(T_1, T_2, S_1, S_2)_i$ obtained by the model spin-up for $f_{1,i}$ to data $m_{d,i}$ from the *Climber2* model [34], an Earth-system model of intermediate complexity. $u = (T_1^*, T_2^*, T_3^*, \Gamma, k_m, a)$ are the control parameters to be optimized. Here, Γ is a thermal coupling constant in the computation of the thermal relaxation constants λ_i, $i = 1, 2, 3$.

With l different freshwater fluxes, the length of the state vector enlarges to $\dim(y) = 8l$ where $y = (y_i)_{i=1}^{l}$ with $y_i = y(f_{1,i}) = (T_{1,i}, T_{2,i}, T_{3,i}, T_{4,i}, S_{1,i}, S_{2,i}, S_{3,i}, S_{4,i})$.

If $F(y, u)$ denotes the right-hand side of the ODE system (3.2), the minimization problem reads

$$\begin{aligned} \min_{y,u} J(y,u), J(y,u) &:= \frac{1}{2}\|m(y(f_1),u) - m_d\|_2^2 + \frac{\alpha}{2}\|u - u_{est}\|_2^2, && (3.3)\\ \text{s.t.} \quad 0 &= (F(y(f_{1,i}),u))_{i=1,\ldots,l}\,. && (3.4) \end{aligned}$$

Applying the full step explicit Euler time stepping, the iteration function to solve $0 = (F(y(f_{1,i}), u))_{i=1,\ldots,l}$, is $G(y, u) = y + F(y, u)$ with $G : Y^l \times \mathbb{R}^6 \to Y^l$.

The difficulty here is that $m : Y^l \times \mathbb{R}^6 \to \mathbb{R}^l$ is not injective. There are several combinations of steady/feasible T_1, T_2, S_1, S_2 and the parameter $u_5 = k$ to compute the same overturning m. The smaller α the more likely the different optimization strategies find completely different optimal parameters with almost the same function values $J(y^*, u^*)$.

Main results concerning the applicability of the One-shot strategy on this model are published in Kratzenstein and Slawig 2013 [22]. The article is attached in the appendix of this work.

Besides the analysis of the theoretical applicability of the One-shot method and a description of the optimization setup, in [22] we investigate the One-shot strategy on the one hand with computation of the preconditioner B as proposed in [10, 11] or summarized in chapter 2.3, equation (2.13), of this work, and on the other hand with the One-shot version using a BFGS approximation to B. For comparison we apply the traditional BFGS quasi-Newton strategy, the limited memory BFGS strategy and the limited memory BFGS strategy with box constraints on the parameters (for description of the methods see for example Nocedal and Wright [25], pp. 136-143 and 176 - 181). The implementation is in FORTRAN. For the computation of derivatives, we apply the

Automatic Differentiation tool TAF [8], version 1.9.52.

3.2.1 Main Results

In the following, we summarize the main results.

- The One-shot strategy computed acceptable results for regularization factors $\alpha > 0$. Figure 3.2 exemplarily illustrates the reduction of the cost function computed by the One-shot strategy and the quasi-Newton BFGS algorithm. We observe the One-shot strategy typically tending to the solution faster than the BFGS strategy comparing the number of Euler steps. For $\alpha = 0$, the One-shot method failed, whereas the BFGS strategy computed optimal parameter values far away from estimated real world values. This gave rise to additionally testing the quasi-Newton strategy with box constraints. We observe the number of Euler iteration steps needed by the One-shot method being smaller than the number of Euler steps needed by the quasi-Newton methods in most cases. However, it was difficult to compare the obtained results as the computed optimal parameters were completely different for the different strategies, especially for $\alpha \leq 0.1$. Furthermore, for the One-shot strategy it is $\|L_{(y,\bar{y},u)}(y^*, \bar{y}^*, u^*)\|$ characterizing the quality of the obtained result $(y^*, \bar{y}^*, u^*)$ whereas it is $\|J_u(y^*(u^*), u^*)\|$ for the quasi-Newton methods. An overview of the obtained results is given in tables 2 and 3 of the attached paper [22].

- Application of the full step explicit Euler iteration on the Rahmstorf 4-box model does not guarantee contraction factors $\rho_k = \|G(y_{k+1}, u) - y_{k+1}\| / \|G(y_k, u) - y_k\|$ with $\rho_k < 1$ for all $k > 0$. In our testings, the contraction rate exceeded 1 up to 16% of the needed Euler steps until a steady state is found for certain pairs of fresh water fluxes f_1, starting values y_0, and parameter sets u. However, the full step explicit Euler time stepping converges and the One-shot method also converges and computes good results. We find that the computed Explicit Euler sequence in our testings fulfills the *quasi-contraction property* [5]

  $$\|y_{k+1} - y_k\| \leq q \max\{\|y_k - y_{k-1}\|, \|y_{k+1} - y_{k-1}\|\}, \text{ where } 0 \leq q < 1, \quad k > 0.$$

  We experience the assumption (2.3) on the spin-up function G, namely $\|G_y\| \leq \rho < 1$ for all $y \in Y$, possibly being defined too strict. It excludes the Euler time stepping applied on the 4-box model even though the spin-up has been found reliable in numerical tests.

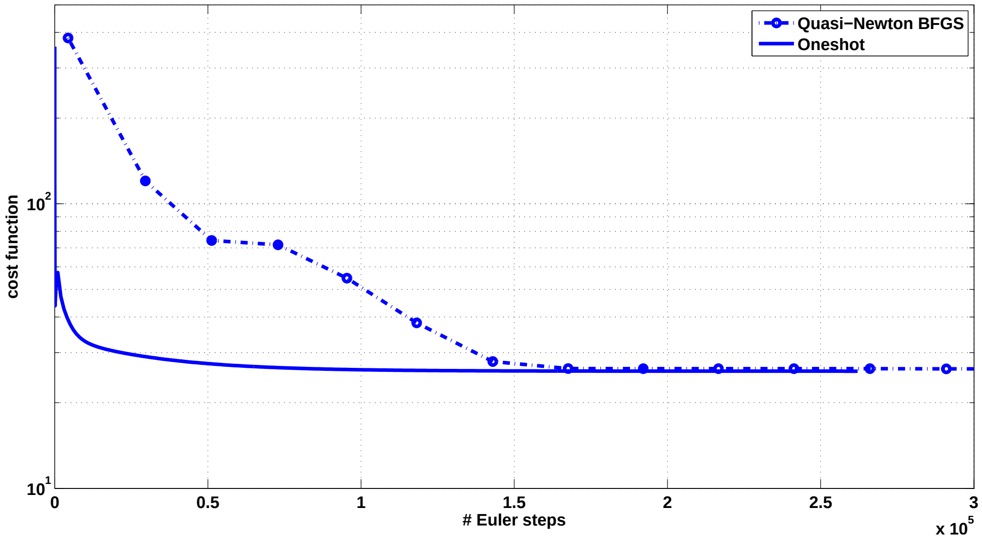

Figure 3.2: Exemplary plot of the reduction of the cost function of problem 3.3 over the number of Euler steps computed by the One-shot strategy and the quasi-Newton BFGS method. Shown is the optimization run for $u \in \mathbb{R}^6$ and weighting factor $\alpha = 10$.

- As the computed ρ exceeds 1 in several iteration steps and would therewith result in incorrectly computed preconditioners B, we fix ρ to $\rho = 0.9$ in our One-shot optimization setup.

- In this example, we observed $\|L_{yy}\|$ computed via the power iteration converging to numbers close to but not equal to 1. However, a simplification in that $\|L_{yy}\|$ and therewith the factors α, β and σ necessary for the computation of the preconditioner B are approximated only every 1,000 One-shot iteration steps is favorable due to an enormous saving of computational time.

- We introduce so-called *pre-iteration* steps in the beginning of the optimization process. These are coupled iteration steps consisting of an update of the state y and the adjoint state $\bar{y}$ only without updating the parameters u. Pre-iterations avoid badly computed parameter sets due to large and possibly immediately reversed corrections of the state y in the beginning of the optimization run. Without pre-iterations we observe the One-shot strategy to occasionally fail depending on chosen starting values. With pre-iterations the One-shot method always converged.

- The One-shot method does not need any constraints on the parameters. The penalty term $\frac{\alpha}{2}\|u - u_{est}\|^2$ of the cost functional fully replaces constraints. As mentioned above, the quasi-Newton strategies require box constraints. For the latter, they are indispensable for small regularization factors α.

- The One-shot method with computation of the preconditioner B formulated in (2.13) does not need any line-search procedure. In the same way as the quasi-Newton methods require line-searches, the One-shot-BFGS strategy also is in need of a line-search. Otherwise it fails. We applied a simple strategy constantly halving the step length until there is a reduction in the cost function.

- The One-shot-BFGS strategy is not as successful as the original version. In most cases it finds a solution, but in our testings needed 1.6 up to 5 times the time the original One-shot version needed until convergence.

- Concerning computational time, for the calibration of this low-complexity model, in our implementation, we observe a saving of time only in cases, where the number of needed One-shot iteration steps is less than half the number of Euler steps needed by the quasi-Newton methods.

3.2.2 Further Results

Complementary results of earlier investigations not published in [22] will be presented in the following. They are collected in preprints on the preprint server of the priority program SPP1253 “Optimization with partial differential equations”, [20, 21].

In very first considerations of the smoothened model (3.2), we optimize parameters $(u_1, u_2, u_3) = (T_1^*, T_2^*, T_3^*)$ only. Therewith, the optimization problem (3.3) is a separable problem where $L_{yu} = 0$.

For these numerical analyses, the implementation is in MATLAB version 7.7.0.471 (R2008b). For comparison with standard optimization strategies, we apply the MATLAB optimization routine `fminunc` with two different settings. Using default `optimoptions` only setting `'LargeScale'='off'` corresponds to the quasi-Newton BFGS algorithm [25], pp. 135 - 143. Additionally setting `'HessUpdate' = 'SteepDesc'` corresponds to the line search method with steepest descent [25], pp. 20-22.

For the computation of the derivatives $\bar{y}^\top G_y(y_k, u_k)$ and $\bar{y}^\top G_{yy}(y_k, u_k)$ for the One-shot method and $J_u(y^*(u_k), u_k)$ for the standard gradient methods, we apply ADiMat version 0.4 [17]. ADiMat uses the source code transformation approach to augment MATLAB codes with routines computing the desired derivatives. In version 0.4 only the forward mode of AD was available. In the current version 0.6, also the reverse mode is

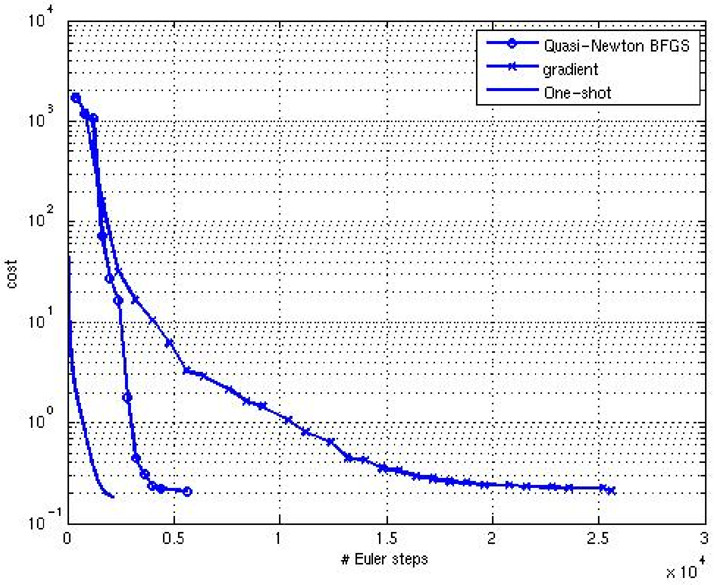

Figure 3.3: Comparison of cost function values over the number of Euler steps in the optimization of 3.3 with three parameters only, $u = (T_1^*, T_2^*, T_3^*)$. Here we use a MATLAB implementation for the One-shot strategy and apply the MATLAB routine `fminunc` with different settings for the comparative optimization strategies, that is the quasi-Newton BFGS and the method of steepest descent.

available.

In the testings with the smoothened version of the box model optimizing three parameters, the required derivatives in the One-shot method with respect to u, i.e. $J_u(y_k, u_k)$, $G_u(y_k, u_k)$ and $L_{uu}(y_k, u_k)$, and additionally $J_{yy}(y_k, u_k)$ can be calculated analytically and be provided by hand written code.

The three methods find optimal solutions. The One-shot strategy converges in only a fraction of needed Euler steps (equal to the number of One-shot steps) compared to the quasi-Newton-BFGS ($\approx 2/5$) and the gradient method ($< 1/10$). See figure 3.3 for illustration.

However, we observe very long computational times which we in great part trace back to the application of ADiMat. On the one hand, it only generates derivative code with the forward mode such that for instance, 81 forward calls are necessary to generate $\bar{y}^\top G_y(y_k, u_k)$. On the other hand, the generated code defines and after usage clears intermediate variables in each Euler step such that a lot of time is spent on the allocation and clearing of variables and memory instead of re-using allocated memory for specific

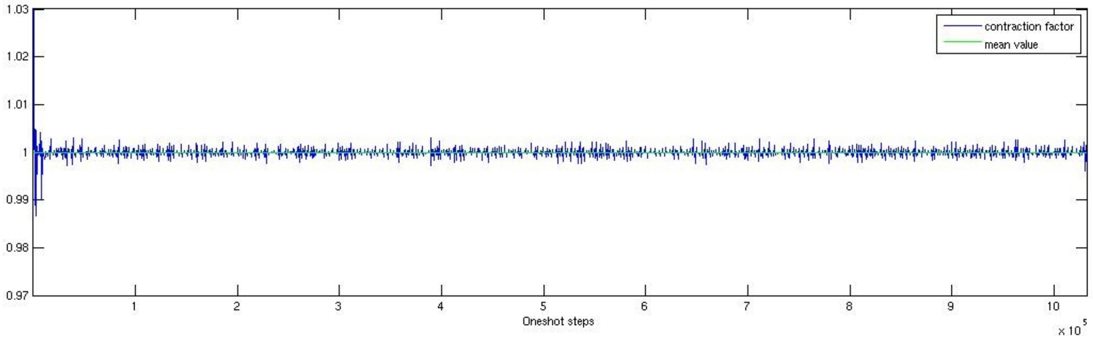

Figure 3.4: Contraction factor ρ_k during an exemplary One-shot optimization run, $\alpha = 0.1$. It exceeds 1 in 4.2 % of all iteration steps. The mean value is $\bar{\rho} = 0.999988$.

variables. We reduced computational time to only a tenth of the originally provided ADiMat code by adjusting the allocation of variables and exploiting the block matrix structure of derivatives with respect to y.

However, the tests still performed very slowly. One coupled One-shot step required 400 times the time of the computation of a single Euler step.

In conclusion, we found that the One-shot strategy successfully operates, but still a well-considered optimization set-up is necessary.

That is why the subsequent tests with the 4-box-model were implemented in FORTRAN and the AD tool TAF [8] was applied to generate derivative code. Similar to ADiMat TAF is a source code transformation tool. The final results are already itemized in the previous subsection and collected in [22].

To complete this chapter, we present the observation concerning the contraction behaviour. As we have experienced the contraction factors $\rho_k > 1$ for some $k > 0$ for certain pairs of (f_1, y_0, u) and observe slow convergence, we do not expect the One-shot strategy to converge fast. However, we are interested in the contraction behaviour. Figure 3.4 illustrates the results. In an exemplary optimization run, we observe $\max_{k \geq 0}(\rho_k) = 1.03 > 1$, and $\rho_k > 1$ in 4.2% of all iteration steps distributed over the entire optimization run. In conclusion, similar to the underlying model-spinup which violates the contraction property (2.4) but still converges consistently in numerical testings, also the One-shot strategy converges in most testings even though the contraction property cannot be proven in general.

As the fitting of *Climber2* data confirmed, the considered box model is suitable for projecting the reaction of the flow of the North Atlantic with respect to climate change.

However, models to simulate parts of the marine ecosystem are becoming more and more complex, see for example Kriest et al. [23] for a comparison of different models. The augmented models take into account additional physical, chemical, biological or biogeochemical details in finer spatial and temporal resolution. Therefore, we take a great interest in the applicability of the One-shot method in the calibration of a high-resolution annually periodic model. The next chapter analyses the One-shot strategy applied to the N-DOP model, a spatially three-dimensional model simulating the phosphorous cycle in the world's oceans.

4 Calibration of the N-DOP Model

After successful testings with the low-dimensional and low-complexity model of the North Atlantic, we will now investigate the One-shot strategy in the parameter optimization of the high-dimensional, annually periodic N-DOP model. Excerpts of the results are published already in the collections [37] and the special volume [2], which collects contributions of the working groups of the DFG priority program 1253 "Optimization with Partial Differential Equations". In this work, we go into detail describing the model, the implementation, and of course results for different optimization set-ups.

4.1 Model Description

In this section, we briefly mention the characteristics of the N-DOP model. Here, we concentrate on its inclusion within the One-shot optimization strategy. In the appendix B, we present a detailed model description. We basically follow the notations of Prieß et al. [32] and Roschat et al. [35, 36].

The N-DOP model describes the global phosphorus cycle in the ocean in an *off-line mode*, meaning that the considered tracers do not affect the ocean physics. The model investigates the two tracers phosphate y_N and dissolved organic phosphorus y_{DOP}. N stands for *nutrients* which in this model is equivalent to the concentration of phosphate PO_4. Thus, we equivalently write N or $PO4$ in the course of this work. The tracers y_N and y_{DOP} are coupled by a coupling function $q(y_N, y_{DOP}, u)$ (also called the *biological source-minus-sink term* or sometimes also *predator-prey-model with sinking*). The ocean circulation data enters the tracer transport equation as annually fixed forcing terms varying during the period of one year. It is precomputed and stored in so called Transport Matrices $A_{imp,j}$ and $A_{exp,j}$ for each intermediate time integration step $j = 0, ..., n_t - 1$, where n_t denotes the number of time steps per year (in our case $n_t = 2{,}880$). The Transport Matrix Method was introduced by Khatiwala et al. [19].

With these fixed linear mappings, finding an annually periodic solution is equivalent to finding a steady state $y = (y_N, y_{DOP})$ at *one* (arbitrary) time point of the year. We define the spin-up function $G(y, u)$ for which we search a state y^* with $y^* = G(y^*, u)$

Table 4.1: List of parameters of interest including the element in the parameter vector, variable name, description, units for the N-DOP model parameters, optimal values in the optimization with synthetic data and starting values in our twin-experiments.

	Name	Description	Unit	u_{opt}	u_0
u_1	λ	remineralization rate of DOP	$1/d$	0.5	0.3
u_2	α	maximum community production rate	$1/d$	2.0	5.0
u_3	σ	fraction of DOP, $\bar{\sigma} = (1 - \sigma)$	-	0.67	0.40
u_4	K_N	half saturation constant of N	$mmolP/m^3$	0.5	0.8
u_5	K_l	half saturation constant of light	W/m^2	30.0	25.0
u_6	K_{H2O}	attenuation of water	$1/m$	0.02	0.04
u_7	b	sinking velocity exponent	-	0.858	0.78

as a concatenation of n_t intermediate time steps:

$$G(y_k, u) = y_{k+1} = y_{k+1,0} := y_{k,n_t} \text{ where} \tag{4.1}$$

$$y_{k,j+1} = A_{imp,j}(A_{exp,j} y_{k,j} + q_j(y_{k,j}, u)), \text{ for } j = 0, ..., n_t - 1. \tag{4.2}$$

q_j is the non-linear coupling function of source-minus-sink type which depends on the parameter and state variables, but it also depends on the intensity of light I which varies in latitude and season. Therefore, we write $q_j(y, u)$ in the evaluation of the model spin-up function G to imply the seasonal dependency.

The parameters of interest and their units are listed in table 4.1. We follow Kriest et al. [23] in the choice of estimated optimal values u_{opt} and for better comparison, we follow Prieß et. al. [32] in the choice of starting values u_0.

In the discretized case, the space domain, denoted $\Omega \subset \mathbb{R}^3$ in the appendix B, consists of 52,749 points in the ocean which are on a 2.8×2.8 grid in 15 vertical layers of different thickness.

4.2 The Optimization Setup

As mentioned above, in this special example finding an annually periodic state is equivalent to finding a steady state y^* at one time point of the year. The length of the state vector y is $\dim(y) = 2 * 52{,}749 = 105{,}498$. In tests with synthetic data, we include data of this single time point only in the cost functional J and $\dim(y_{data}) = \dim(y)$. The cost functional reads

$$J(y, u) = \frac{1}{2} \|y - y_{data}\|_2^2 + \frac{\alpha}{2} \|u - u_{est}\|_2^2 . \tag{4.3}$$

The Euclidean norm is employed without any weighting as for the state y the discretization with different thicknesses of vertical ocean layers introduces a certain weighting in y. Weights in the parameters can easily be included if desired.
The inclusion of more than one time point in the optimization problem is described in section 4.5.
The model spin-up $G(y_k, u) = y_{k+1}(u) \overset{k\to\infty}{\longrightarrow} y^*(u)$ is performed as described in section 4.1. One step $y_{k+1} = G(y_k, u_k)$ consists of $n_t = 2{,}880$ intermediate time steps per model year. In [23], the authors regard a spin-up of 3,000 model years being converged. We find that mathematically and in the optimization context a stricter condition is needed as after 3,000 model years, the $\|\cdot\|_2$ norm of the change in the state still is $1.6 * 10^{-4}$. We observe a spin-up being converged with a required accuracy of less than 10^{-5} after approximately 10,000 model years.

For the computation of the needed derivatives $\bar{y}^\top G_y$, $\bar{y}^\top G_u$, $\bar{y}^\top G_{yu}$, $\bar{y}^\top G_{uu}$ it is essential finding an efficient strategy. Even though the required derivatives depend on the current iterates $(y_k, \bar{y}_k, u_k)$ only, using finite differences would mean to perform those 2,880 intermediate time steps n and m times for G_y and G_u respectively in each iteration step, which leads to high computational costs. As described in chapter 2.4, a cheaper, faster and most importantly exact strategy with no approximation errors is to apply a tool for automatic (or algorithmic) differentiation (AD).

4.2.1 Automatic Differentiation for the N-DOP Model

For numerical tests with the N-DOP model, we apply the AD tool TAF but also compute derivatives analytically (see table 4.2 for an overview). In particular, we compute derivative code of the coupling term q (see the appendix (B.6),(B.7)) with TAF, and concatenate it with the transport matrices $A_{imp,j}$ and $A_{exp,j}$, $j = 0, ..., n_t - 1$, resulting in AD based codes for the reverse and the forward mode. In Piwonski's *Metos3d* context [31], a software toolkit for optimization and simulation of marine ecosystem models, this enables us to easily include a different coupling function q for a different model.
Exemplarily, the pattern for the computation of $\bar{y}_k^\top G_y(y_k, u_k)$ is given in algorithm 1 of figure 4.1. The pattern for the computation of $\bar{y}_k^\top G_u(y_k, u_k)$ is analogous replacing $\frac{\partial q_j(y_{k,j}, u_k)}{\partial y_{k,j}}$ by $\frac{\partial q_j(y_{k,j}, u_k)}{\partial u_k}$. In the implementation, $\bar{y}_k^\top G_u(y_k, u_k)$ and $\bar{y}_k^\top G_y(y_k, u_k)$ are computed in one call.
In the reverse mode, applying AD on the *entire* spin-up with $n_t = 2{,}880$ intermediate time steps would include also storing the intermediate values of the computation of

$q_j(y_{k,j}, u_k)$ demanding huge amount of storage. Instead, we apply so-called checkpointing: we store the intermediate values $y_{k,j}$, $j = 0, ..., n_t - 1$, and, in the reverse sweep, recompute and only then store intermediate values of $q_j(y_{k,j}, u_k)$ for the computation of $\bar{z}_{i_1}^\top \frac{\partial q_j(y_{k,j},u_k)}{\partial y}$ and $\bar{z}_{i_2}^\top \frac{\partial q_j(y_{k,j},u_k)}{\partial u}$, where $\bar{z}_{i_1}$, $\bar{z}_{i_2}$ are intermediate values in the computation of the desired vector-Jacobian products.

For the computation of $G_u(y_k, u_k)$ the forward mode is favorable due to the small dimension of u. The pattern of the forward mode of AD for the N-DOP model is presented in algorithm 2 of figure 4.1.

We observe high sensitivity of the spin-up function G with respect to the parameters which results in considerable differences in the Jacobian G_u computed by the approaches AD and Finite Differences. For the Finite Differences approach (FD) where for some small scalar ϵ the i-th column of G_u is approximated by

$$\frac{\partial G(y,u)}{\partial u_i} = \frac{G(y, u + \epsilon e_i) - G(y,u)}{\epsilon} \tag{4.4}$$

we already observe large differences for different choices of ϵ, especially in the 6-th column which is the derivative of G with respect to $u_6 = K_{H2O}$. For $\epsilon_1 = 10^{-8}$ and $\epsilon_2 = \sqrt{\epsilon_1 * (1 + \|u_k\|)}/1 \approx 5.56 \cdot 10^{-8}$ (as proposed in [28]) the euclidean norm of the difference of the two computed $\frac{\partial G(y,u)}{\partial u_6}$ was ≈ 20.54. The largest differences are detected at the top layer for both tracers. The comparison of FD with ϵ_2 and AD was even more significant. The $\| \cdot \|_2$-norm of the difference of the computed Jacobians in the 6th column was 220.15. With values up to 6.6, the largest differences again are observable in the uppermost ocean layer where $\frac{\partial G(y,u)}{\partial u_6}$ contains values in the lower 2-digit range. In the other columns, the norm of the differences was between 8.3 and 23.5.

However, since we need the full Jacobian G_u in the approximation of the preconditioner B only, which itself is an approximation of the doubly augmented Lagrangian, we do not want to dogmatically insist on the application of AD in this special case of the computation of G_u. The strategy performed almost equally in our testings for both approaches in the computation of G_u.

The same holds for the second order derivatives $\bar{y}^\top G_{yu}$ and $\bar{y}^\top G_{uu}$, for which we applied the FD approach after AD computation of $\bar{y}^\top G_y$ and $\bar{y}^\top G_u$ respectively.

4.2.2 Simplifications

For the computation of the weights σ, α_L and β_L of the preconditioner B defined in (2.13), we choose the original version defined in (2.9) and (2.11) due to avoiding the

Algorithm 1: AD Computation of $\bar{y}_k^\top G_y(y_k, u_k)$.

```
Input: $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^n$ , $\bar{y}_k \in \mathbb{R}^n$

$y_{k,0} = y_k$
for j=0, ..., $n_t - 1$ do                                       // Forward loop
    $z_0 = y_{k,j}$
    $z_1 = q_j(z_0, u_k)$
    $z_2 = A_{exp,j} z_0$
    $z_3 = z_1 + z_2$
    $z_4 = A_{imp,j} z_3$
    "Checkpointing":  store  $y_{k,j+1} = z_4$
end
Output 1:  $G(y_k, u_k) = y_{k,n_t}$

$\bar{y}_{k,n_t} = \bar{y}_k$
for j = $n_t - 1, \ldots, 0$ do                                  // Reverse loop
    $\bar{z}_4 = \bar{y}_{k,j+1}^\top$
    $\bar{z}_3 = \bar{z}_4 A_{imp,j}$
    $\bar{z}_2 = \bar{z}_3$
    $\bar{z}_1 = \bar{z}_3$
    $\bar{z}_0 = \bar{z}_2 A_{exp,j}$
    call TAF generated routine to provide $\bar{z}_1 \frac{\partial q_j(y_{k,j}, u_k)}{\partial y_{k,j}}$
    $\bar{z}_0 = \bar{z}_0 + \bar{z}_1 \frac{\partial q_j(y_{k,j}, u_k)}{\partial y_{k,j}}$
    $\bar{y}_{k,j} = \bar{z}_0^\top$
end
Output 2:  $\bar{y}_k^\top G_y(y_k, u_k) = \bar{y}_{k,0}$
```

Algorithm 2: AD Computation of $G_u(y_k, u_k)$.

```
Input: $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^n$, $I_m = (e_i)_{i=1,\ldots,m}$

$y_{k,0} = y_k$
for i=1,...,m do
    $\dot{y}_{k,0,i} = 0$
end
for j=0, ..., $n_t - 1$ do                                       // Forward loop only
    $z_0 = y_{k,j}$
    for i=1,...,m do
        $\dot{z}_{0,i} = \dot{y}_{k,j,i}$
        call TAF generated routine to provide $z_1 = q_j(z_0, u_k)$ and $\dot{z}_{1,i} = \frac{\partial q_j(z_0,u_k)}{\partial y} \dot{z}_{0,i} + \frac{\partial q_j(z_0,u_k)}{\partial u} e_i$
    end
    $z_2 = A_{exp,j} z_0$
    for i=1,...,m do
        $\dot{z}_{2,i} = A_{exp,j} \dot{z}_{0,i}$
    end
    $z_3 = z_1 + z_2$
    for i=1,...,m do
        $\dot{z}_{3,i} = \dot{z}_{1,i} + \dot{z}_{2,i}$
    end
    $z_4 = A_{imp,j} z_3$
    for i=1,...,m do
        $\dot{z}_{4,i} = A_{imp,j} \dot{z}_{3,i}$
    end
    $y_{k,j+1} = z_4$
    for i=1,...,m do
        $\dot{y}_{k,j+1,i} = \dot{z}_{4,i}$
    end
end
Output:  $G(y_k, u_k) = y_{k,n_t}$, $G_u(y_k, u_k) = (\dot{y}_{k,n_t,i})_{i=1,\ldots,m}$
```

Figure 4.1: Reverse and forward mode of AD for the spin-up function G.

Table 4.2: Computation of derivatives using the approaches of Automatic Differentiation (AD) and Finite Differences (FD).

Derivative	Mode of Computation
J_y, J_u	analytically; in chapter 4.5 AD reverse mode
$\bar{y}^\top G_y$, $\bar{y}^\top G_u$	one reverse sweep of AD + analytically for linear parts
G_u	forward mode of AD + analytically for linear parts
J_{yu}, J_{uu}	analytically; in chapter 4.5 FD over AD reverse computation of J_y and J_u
$\bar{y}^\top G_{yu}$, $\bar{y}^\top G_{uu}$	FD over AD computation of $\bar{y}^\top G_y$, $\bar{y}^\top G_u$

introduction of a line search procedure. We fix $\|L_{yy}\| = 1$. Due to the lack of knowledge of the contraction factor ρ and its occasional exceeding of 1 in our testing with the model spin-up, we fix it to a number less than but close to 1. In contrast to the Atlantic box model considered in chapter 3, we are going to experience that the choice of ρ is crucial for the functioning of the One-shot strategy in the calibration of the N-DOP model.

During the optimization process, we observe α_L, β_L and σ from (2.11) being almost equal to α_L, β_L and σ from (2.12) even though we are considering a non-separable problem. However, we state that $\|G_u\| \gg \|L_{yu}\|$. such that $\frac{\|L_{yu}\|}{\|G_u\|}$ is very small. That is why in most testings we apply the cheaply computable version of α_L, β_L and σ from (2.12). We keep the option open to adjust σ to compensate for a line search.

To ensure admissible preconditioners B and wise parameter corrections especially in the beginning of the optimization process, we perform *pre-iterations* before starting One-shot iteration steps. Pre-iterations consist of an update of the state and the adjoint state with fixed (starting) parameter values. In our testings, in which we initialize the tracer concentrations at all points of Ω equally ($(y_{N,0})_i = 2.17$, $(y_{DOP,0})_i = 0.0001$, $i = 1, ..., \dim(y_N)$), 500 pre-iterations (compared to 10,000 steps for a full model spin-up) proved to be more than adequate.

Due to only very small changes in the preconditioner B compared to high computational costs, we propose to update the preconditioner only every 5 iteration steps when the change in the parameters and the state are small. In our testings, after 500 so-called *exact One-shot steps* with full computation of B in each step, we use the same pre-conditioner B in five successive steps. We call those *inexact One-shot steps.* They demand less than a quarter of the computational time of exact One-shot steps (see section 4.3.4 for details). The difference in the tracer outputs after 500 exact compared to 500 inexact One-shot steps was insignificant. The same holds for the computed parameter sets.

Note that these inexact One-shot steps are not to be mistaken with Bosse's Multi-step

Algorithm 3: The One-shot Optimization Strategy for the N-DOP Model

```
Initialize: $u_0 \in \mathbb{R}^m$, $y_0 \in \mathbb{R}^n$ , $\bar{y}_0 \in \mathbb{R}^n$, fix $\rho$ to $\rho = 0.9$          // Initialization

for k=0 to 499 do                                                       // Pre-Iteration
    $y_{k+1} = G(y_k, u_k)$
    $\bar{y}_{k+1} = \bar{y}_k^T G_y(y_k, u_k) + J_y(y_k, u_k)$
    $u_{k+1} = u_k$
end

for k=500 to 999 do                                                // Exact One-shot steps
    $y_{k+1} = G(y_k, u_k)$
    $\bar{y}_{k+1} = \bar{y}_k^T G_y(y_k, u_k) + J_y(y_k, u_k)$
    Compute B
    $u_{k+1} = u_k - B^{-1}(L_u(y_k, \bar{y}_k, u_k))$
end

while convergence criteria are not fulfilled do                  // Inexact One-shot steps
    $y_{k+1} = G(y_k, u_k)$
    $\bar{y}_{k+1} = \bar{y}_k^T G_y(y_k, u_k) + J_y(y_k, u_k)$
    Compute B every 5 steps only
    $u_{k+1} = u_k - B^{-1}(L_u(y_k, \bar{y}_k, u_k))$
    $k = k + 1$
end
```

Figure 4.2: The One-shot algorithm as we have implemented it for the N-DOP model.

One-shot steps [2], where after a parameter update multiple state followed by multiple adjoint state updates are performed. The inexact One-shot steps still correct parameters in each iteration step. The preconditioner B remains the same whereas L_u is newly computed in each step such that the parameter correction is not the same during the 5 successive inexact steps.

An overview of our implementation of the One-shot strategy can be found in figure 4.2. The code was implemented by Jaroslaw Piwonski in the PetSC framework [1] for parallel computing. It can easily be included in Piwonski's *Metos3d* toolkit [31].

4.3 Reproducing Synthetic Data

First tests pursue the goal to reproduce model parameters and data in a *twin experiment* framework. The data to be fit is obtained from the N-DOP model spin-up after 10,000 model years with the parameter set u_{opt} from table 4.1. The benefit of such an experiment is the knowledge of optimal parameters, the knowledge of all components of the solution of the state y^* and that those can be reached by the model. The change in the state $\|y_{k+1} - y_k\|_2$ after 10,000 simulation years is about $0.9 * 10^{-5}$. The change in the state does not become smaller during the next 20,000 model years and we suspect that minimal changes are only due to roundoff errors in machine precision.

We briefly mention that One-shot optimization tests with coarser time steps, which Prieß et al. [33] apply, failed. We find that the change in the annual cycle of two consecutive models years was not small enough (10^{-4} and even larger).

Before we get into detail, we concede that the optimization of all 7 considered parameters does not perform optimally with an uncautiously chosen optimization setup. In contrast to that, the optimization of 6 parameters only, fixing the velocity exponent parameter $u_7 = b$ to $u_7 = 0.858$, performs very well with an optimization setup directly taken over from experiments with the Atlantic box model from section 3.
That is why we summarize general results and characteristic behaviors of the strategy obtained for the optimization of 6 parameters first (sections 4.3.1 and 4.3.2), and provide an extra section 4.3.3 for the case of 7 considered parameters describing the observations and possible proposals for the solution of the observed problems.

4.3.1 Results for different weighting factors

Numerical tests with 6 parameters, fixing u_7 to $u_7 = 0.858$ perform very well. We set the contraction rate ρ to $\rho = 0.9$. The description of the results in this section applies to data at January 1st, 0:00 a.m. We obtained almost identical results choosing synthetic data at April 1st, July 1st and October 1st, each at 0:00 a.m.

First, we investigate the performance of the One-shot method for $u_{est} = u_{opt}$. Not surprisingly, we find that the larger α the better the data is fit and the cost function is significantly reduced already after 15,000 One-shot iteration steps and even more after 20,000 steps, see figure 4.3. However, we observe that there is no improvement of the fit after 18,000 steps for $\alpha = 100$ even though the parameter values are still corrected. Figure 4.4 illustrates the reduction of the two parts of the cost functional $\|y - y_{data}\|_2$ and $\|u - u_{est}\|_2$.
We present surface plots of the obtained results in the upper, euphotic zone in figure 4.5 and in the deeper, non-euphotic zone in figure 4.6. Note that shown plots are representative meaning that the quality of the results in other depth layers of the euphotic zone and non-euphotic zone and of the second tracer DOP are similar. We clearly see differences to data for $\alpha = 1$ and $\alpha = 0$ in the upper layers of the ocean, the euphotic zone, whereas there is no difference to be detected for $\alpha = 100$ in the euphotic and for all α in the non-euphotic zone already after 15,000 One-shot steps.
In figure 4.7 we illustrate parameter values during the optimization process. They all almost hit optimal values.
An overview of the obtained cost function values, parameter sets and the euclidean

Table 4.3: Analysis of optimality conditions. A KKT point fulfills $\|\nabla L\| = \|L_{(y,\bar{y},u)}\| = 0$. To further illustrate convergence behaviour, we list the change in the parameters $\|u_{k+1} - u_k\|_2$. The listed factors are the quotients $\|L_y(y_0, \bar{y}_0, u_0)\| / \|L_y(y_k, \bar{y}_k, u_k)\|$ and $\|L_u(y_0, \bar{y}_0, u_0)\| / \|L_u(y_k, \bar{y}_k, u_k)\|$ respectively.

	$\|L_{\bar{y}}(y_k, \bar{y}_k, u_k)\|_2$	$\|L_y(y_k, \bar{y}_k, u_k)\|_2$	factor	$\|L_u(y_k, \bar{y}_k, u_k)\|_2$	factor	$\|u_{k+1} - u_k\|_2$
k = 15,000:						
$\alpha = 100$	$5.1 * 10^{-5}$	$8.4 * 10^{-2}$	1,964	$1.062 * 10^{2}$	690	$2.5 * 10^{-6}$
$\alpha = 1$	$1.6 * 10^{-4}$	$5.6 * 10^{-2}$	2,946	$2.919 * 10^{2}$	251	$2.3 * 10^{-5}$
$\alpha = 0$	$1.6 * 10^{-4}$	$5.7 * 10^{-2}$	2,894	$2.934 * 10^{2}$	250	$1.2 * 10^{-4}$
k = 20,000:						
$\alpha = 100$	$1.3 * 10^{-5}$	$8.3 * 10^{-2}$	1,987	$2.699 * 10^{1}$	2,715	$4.9 * 10^{-7}$
$\alpha = 1$	$6.5 * 10^{-5}$	$5.6 * 10^{-2}$	2,946	$1.188 * 10^{2}$	617	$9.5 * 10^{-6}$
$\alpha = 0$	$6.8 * 10^{-5}$	$5.7 * 10^{-2}$	2,894	$1.222 * 10^{2}$	600	$5.7 * 10^{-5}$

norm of L_u, as one indicator of the quality of the results, is given in table 4.4.

For further examination of the quality of the obtained results, we consider the gradient of the Lagrangian $\nabla L = L_{y,\bar{y},u} = (L_y, L_{\bar{y}}, L_u)$ in table 4.3, which must satisfy $\nabla L = 0$ at a KKT point, and additionally the norm of the step in the parameters $\|u_{k+1} - u_k\|_2$. Note that $\|L_{\bar{y}}\|_2$ corresponds to the change in the state variables, $\|L_y\|_2$ to the change in the adjoint states. As $\|L_y\|$ and $\|L_u\|$ are not (numerically) zero after 20,000 iteration steps, we list factors representing the rates of the reduction of $\|L_y\|$ and $\|L_u\|$ computed from $\|L_y(y_0, \bar{y}_0, u_0)\| / \|L_y(y_k, \bar{y}_k, u_k)\|$ and $\|L_u(y_0, \bar{y}_0, u_0)\| / \|L_u(y_k, \bar{y}_k, u_k)\|$ respectively.

We can conclude that concerning the cost functional, good results are already obtained after 15,000 One-shot iteration steps. However, concerning the optimality, even after 20,000 steps the computed vectors are far away from the optimality condition $\nabla L = 0$. Results obtained after 20,000 steps for $\alpha \in \{1, 0\}$ are as good as those obtained after 15,000 steps for $\alpha = 100$. So, for all considered weights α, the optimization results in the same optimal parameters and state variables but with slower convergence the smaller α is chosen.

Surprisingly, good results are obtained even if $\alpha = 0$ and therewith without any penalty term in the cost functional.

A closer look at the preconditioner B and the gradient of the Lagrangian w.r.t. u, L_u, reveals that in this special example of the N-DOP model considered with 6 parameters and with one time point only at which data is given, the weighting factor α does not significantly influence the performance of the One-shot strategy if it is chosen $\alpha \leq 100$.

B was defined as

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L (J_{yu} + \bar{y}^\top G_{yu})^\top (J_{yu} + \bar{y}^\top G_{yu}) + J_{uu} + \bar{y}^\top G_{uu}),$$

which is in this case, where $J_{yu} = 0$,

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L (\bar{y}^\top G_{yu})^\top (\bar{y}^\top G_{yu}) + \alpha I + \bar{y}^\top G_{uu}).$$

The step $s_k = u_{k+1} - u_k$ is computed from

$$\begin{aligned} & B_k s_k &=& -L_u(y_k, \bar{y}_k, u_k) \\ \Leftrightarrow\quad & B_k s_k &=& -(\alpha(u_k - u_{est}) + \bar{y}_k^\top G_u(y_k, u_k)). \end{aligned}$$

Choosing $\rho = 0.9$, the weighting factors equal $\sigma = 0.05$, $\alpha_L = 400$, $\beta_L = 2$. At the beginning of the optimization, the row-sum norms $\|\cdot\|_1$ of the symmetric matrices have the orders of magnitude $\|G_u^\top G_u\|_1 \approx 10^4$, $\|(\bar{y}^\top G_{yu})^\top (\bar{y}^\top G_{yu})\|_1 \approx 10^3$ and $\|\bar{y}^\top G_{uu}\|_1 \approx 10^5$. In the gradient L_u, there too, it is $\bar{y}^\top G_u$ with entries at a dimension of 10^5 that dominates the term. Whereas $\|(\bar{y}^\top G_{yu})^\top (\bar{y}^\top G_{yu})\|_1$ and $\|L_u\|_2$ tend to zero during the optimization process, the dominating terms remain $G_u^\top G_u$ and $\bar{y}^\top G_{uu}$ at the same magnitude as in the beginning of the optimization process such that the weighting factor α in front of the penalty term $\frac{1}{2}\|u - u_{est}\|_2^2$ in the cost functional plays a minor role during the optimization process in this special example if it is chosen comparatively small as done in our testings with $\alpha \leq 100$.

4.3.2 Twin Experiment with non-optimal $\mathbf{u_{est}}$

In this section, our attention is drawn on the case which is closer to real world optimization problems where optimal parameters are not known. Now, we consider $u_{est} \neq u_{opt}$ such that the cost functional cannot become zero. We chose $\alpha = 0.01$ to only very slightly keep parameters near u_{est}.

As we have analyzed previously, in this example the penalty term is insignificant if α is chosen small. Thus we expected the One-shot strategy to perform very well.

The reduction of the cost functional and the parameter values during the optimization run are illustrated in figures 4.8, 4.9 and table 4.5. Figure 4.8 displays the cost function and its parts not only over the number of One-shot iteration steps but also over the equivalent number of model spin-ups for later comparison. See the section on computational time, section 4.3.4, for details.

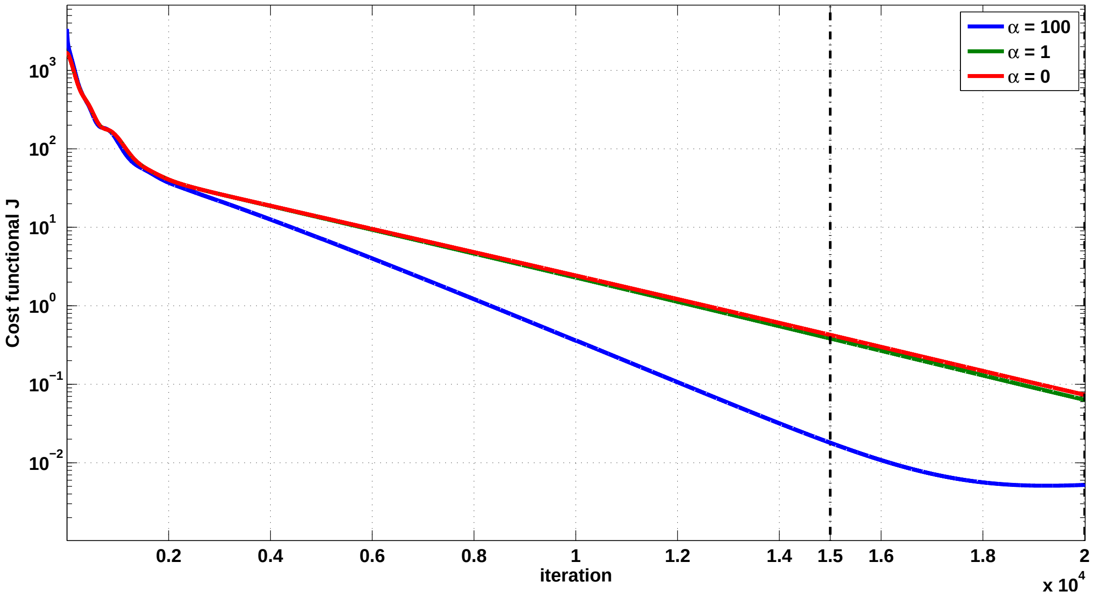

Figure 4.3: Reduction of the cost functional for different α, $u_{est} = u_{opt}$.

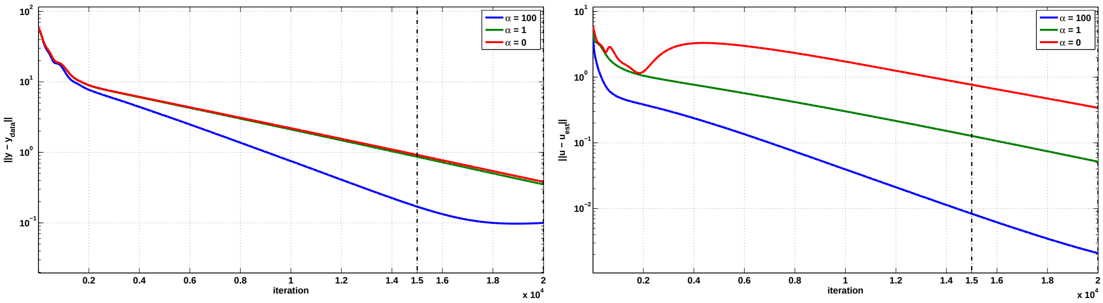

Figure 4.4: Results of the optimization for different α, $u_{est} = u_{opt}$. Left $\|y - y_{data}\|$, right $\|u - u_{est}\|$.

Table 4.4: Overview of initial and optimal parameter sets, the function values at initial parameter and state variables, computed function value, data fit and parameter values after 15,000 and 20,000 One-shot steps for the case where $u_{opt} = u_{est}$. Furthermore, we list the euclidean norm of the gradient of the Lagrangian, L_u, at the respective iteration steps.

	$J(y_0,u_0)$	$J(y^*,u^*)$	fit	u_1	u_2	u_3	u_4	u_5	u_6	$\|L_u\|_2$
u_0				0.3	5.0	0.4	0.8	25.0	0.04	
u_{opt}				0.5	2.0	0.67	0.5	30.0	0.02	
15,000 One-shot steps:										
α=100	3,372.7	0.018	0.17	0.4990	2.005	0.6697	0.4937	29.99	0.0202	$1.062*10^2$
α=1	1,679.7	0.383	0.86	0.5203	2.125	0.6761	0.4902	29.99	0.0212	$2.919*10^2$
α=0	1,662.6	0.423	0.92	0.5222	2.114	0.6766	0.4901	29.24	0.0214	$2.934*10^2$
20,000 One-shot steps:										
α=100		0.005	0.10	0.5000	2.001	0.6700	0.4983	30.00	0.0200	$2.699*10^1$
α=1		0.063	0.35	0.5087	2.051	0.6726	0.4960	29.99	0.0205	$1.188*10^2$
α=0		0.073	0.38	0.5096	2.047	0.6729	0.4958	29.66	0.0206	$1.222*10^2$

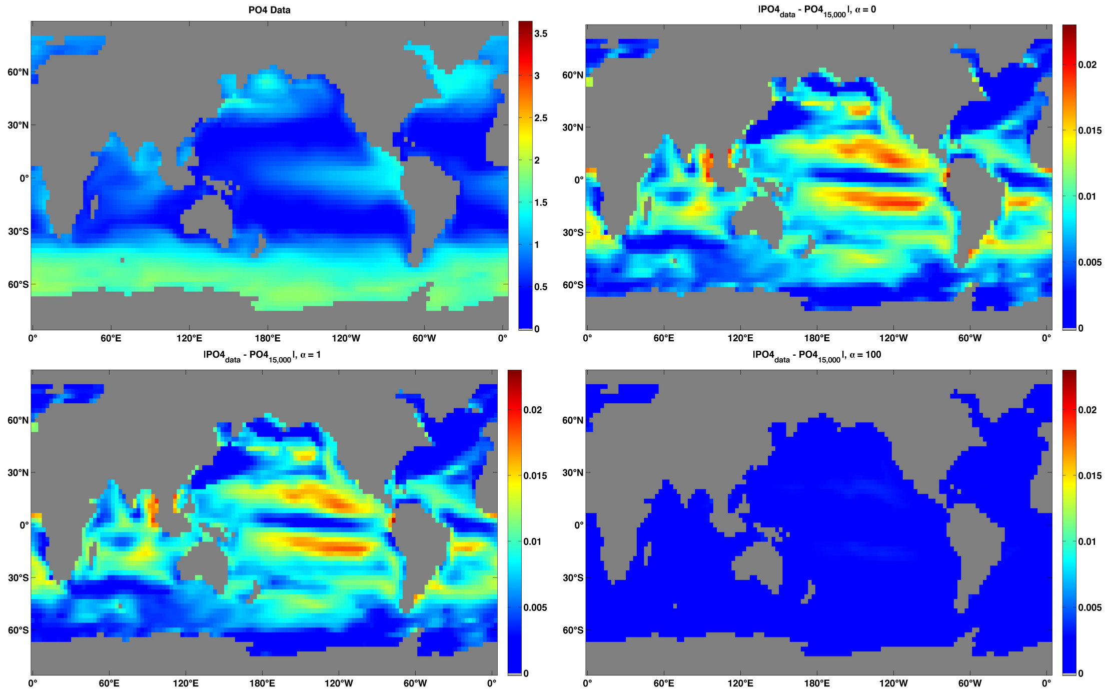
PO4 Data
|PO4data - PO4 15,000|, α = 0
|PO4data - PO4 15,000|, α = 1
|PO4data - PO4 15,000|, α = 100
60°N
30°N
0°
30°S
60°S
0°
60°E
120°E
180°E
120°W
60°W
0°
3.5
3
2.5
2
1.5
1
0.5
0
0.02
0.015
0.01
0.005
0

Figure 4.5: Plots for the concentration of PO4 (equivalent to N) in the first/upper most layer (0 - 50m b.s.l.). Top left: PO4 data, top right: difference to data for $\alpha = 0$, bottom left: difference to data for $\alpha = 1$, bottom right: difference to data for $\alpha = 100$, after 15,000 One-shot steps.

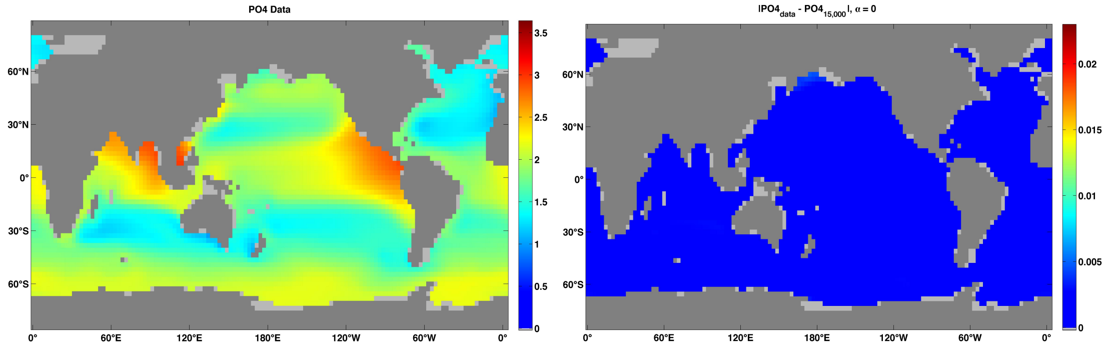

Figure 4.6: Left: Concentration of PO4 (equivalent to N) in the fifth depth layer (360 - 550m b.s.l.) , Right: Difference to data in this depth for $\alpha = 0$, after 15,000 One-shot steps.

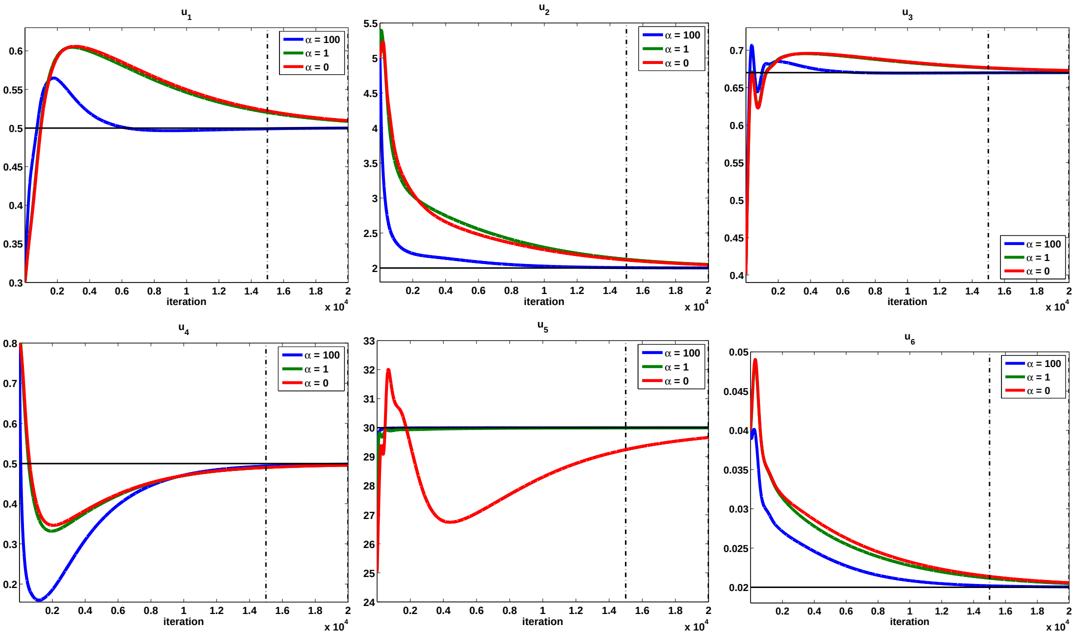

Figure 4.7: Parameter values during the optimization for different α, $u_{est} = u_{opt}$.

Figure 4.9 shows parameter values during the optimization process. We find that in this example 20,000 steps are necessary to obtain mathematically acceptable results. The comparison of the results obtained after 15,000 and 20,000 iteration steps is listed in table 4.5. Compared to tests from section 4.3.1, the calculated value of the fit after 20,000 iteration steps almost corresponds to the value obtained after 20,000 iteration steps for $\alpha = 0$ even though the half saturation constant of light $K_l = u_5$ differs significantly from $u_{opt,5} = 30$ and stays closer to $u_{est,5} = 27$. In both tests, the worst fit is obtained on the top layer, but due to the significantly different parameter u_5 the regions where the fit is bad differ. See figure 4.10 for illustration of the differences exemplarily in the top ocean layer and in the fifth layer. However, compared to tests with real world data in the subsequent section 4.4, the differences to data are detectable on a very small scale only.

4.3.3 Optimization of 7 Model Parameters and the Influence of the sinking velocity exponent u$_7$

One-shot optimization runs with 7 model parameters do not perform as successfully as those with 6 parameters only. Using the same test setting as in the tests with 6 param-

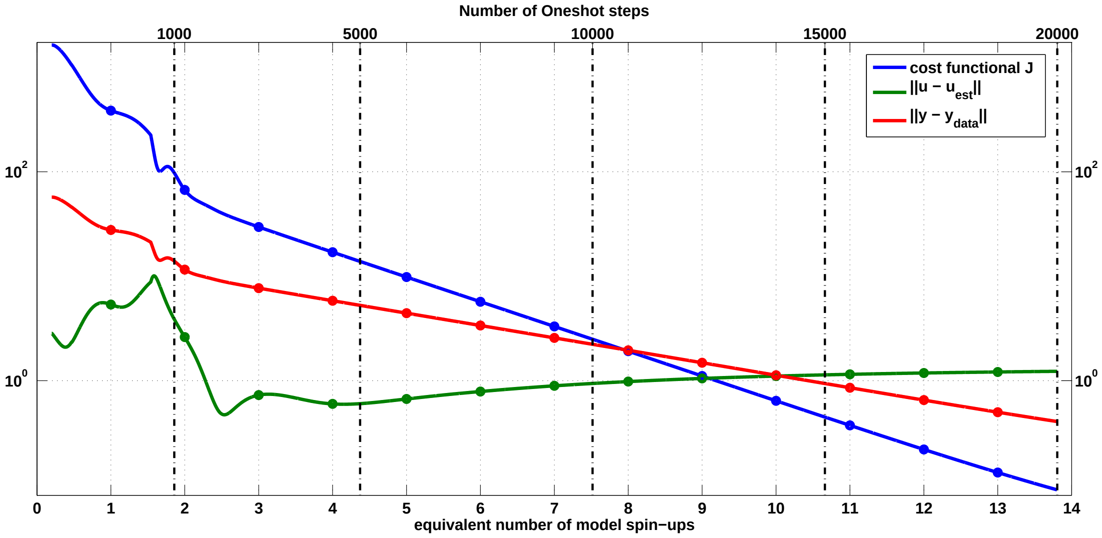

Figure 4.8: Reduction of the cost functional and its parts over the equivalent number of model spin-ups (10.000 model years) converted from computational time for the case where $u_{est} \neq u_{opt}$ and $\alpha = 0.01$.

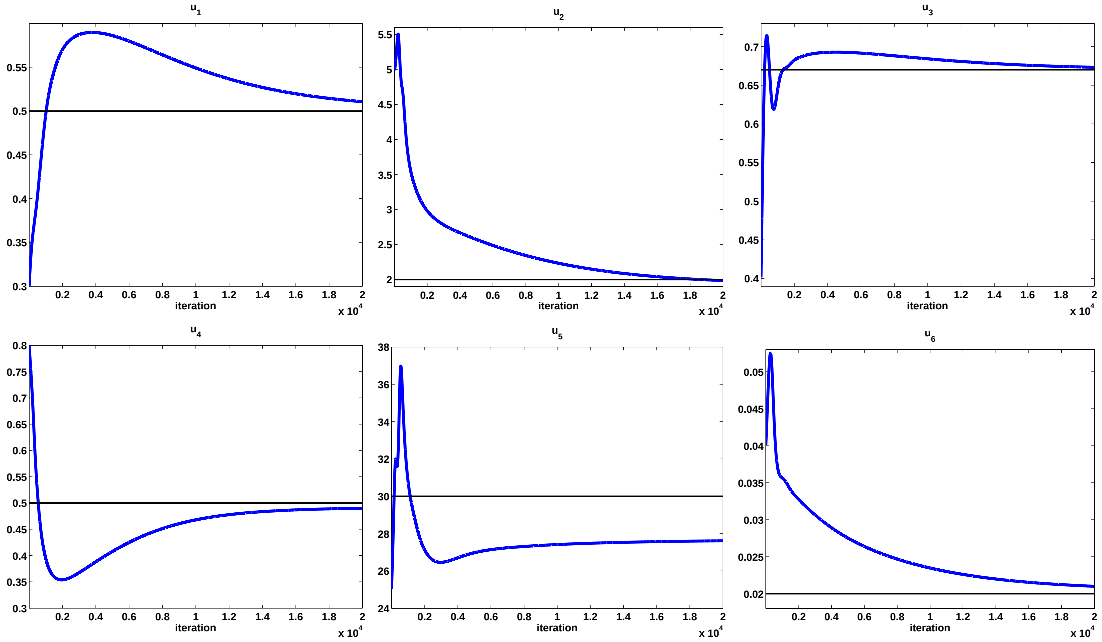

Figure 4.9: Parameter values during the optimization for $u_{est} \neq u_{opt}$, $\alpha = 0.01$.

Table 4.5: Computed cost functional and parameter values after 15,000 and 20,000 One-shot steps for the case where $u_{est} \neq u_{opt}$, $\alpha = 0.01$.

	$J(y_0, u_0)$	$J(y^*, u^*)$	fit	u_1	u_2	u_3	u_4	u_5	u_6	$\|L_u\|$
u_{opt}				0.5	2.0	0.67	0.5	30.0	0.02	
u_{est}				0.45	3.0	0.6	0.8	27.0	0.22	
15,000 steps	1,636.7	0.45	0.939	0.5230	2.06	0.6769	0.4854	27.55	0.02174	$2.92 * 10^2$
20,000 steps		0.0899	0.406	0.5106	1.98	0.6732	0.4899	27.61	0.02100	$1.20 * 10^2$

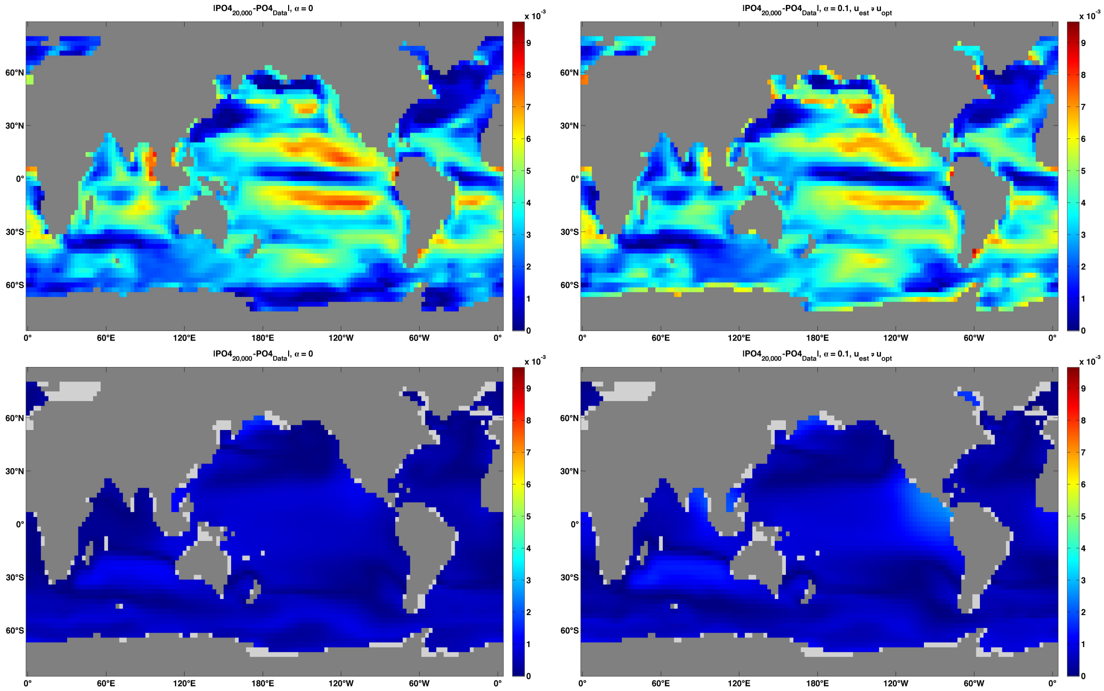
$|PO4_{20,000}-PO4_{Data}|$, $\alpha = 0$
$|PO4_{20,000}-PO4_{Data}|$, $\alpha = 0.1$, $u_{est} \neq u_{opt}$
$|PO4_{20,000}-PO4_{Data}|$, $\alpha = 0$
$|PO4_{20,000}-PO4_{Data}|$, $\alpha = 0.1$, $u_{est} \neq u_{opt}$
$\times 10^{-3}$
9
8
7
6
5
4
3
2
1
0
60°N
30°N
0°
30°S
60°S
0°
60°E
120°E
180°E
120°W
60°W
0°

Figure 4.10: Misfit of the concentrations of PO4 (equivalent to N) after 20,000 One-shot iteration steps for $\alpha = 0$ (left) and for $\alpha = 0.1$ with non-optimal u_{est}. Top: Uppermost layer (0-50m b.s.l.), bottom: Fifth layer (360-550m b.s.l.).

eters, the parameters and therewith the tracer concentrations result in an oscillating movement and the cost functional is not reduced anymore after approximately 5,000 iteration steps. See figure 4.11 for illustration. Parameter values during the optimization run are shown exemplarily for parameters u_1, u_6 and u_7 in figure 4.12.

Choosing different starting values for parameters, state and adjoint state lead to almost identical results. Tests with $\alpha = 1$ and $\alpha = 0$ result in the same oscillating outputs. We observe similar results concerning the oscillation, i.e. oscillation between similar extremes and with a similar period. Also tests with a different date, namely July 1st, 0:00 a.m., perform identically.

Choosing the original factors σ, β_L and α_L defined in (2.9) and (2.11) instead of the simplified versions from (2.12) does not improve the progress of the optimization.

We observe the values of L_u oscillating without any damping. Due to the permanent change in the parameters during the optimization run, also the state y is far away from convergence to a steady annual cycle. Comparing to results with only 6 parameters, we state that the norm of the steps in the parameters is 2 to 3 orders of magnitude larger than in the tests with six parameters. Instead of an order of 10^{-5}, the norm of the steps lies between the orders of magnitude 10^{-3} and 10^{-2}.

We observe that the entries of the preconditioner B and its parts $(\bar{y}^\top G_{yu})^\top(\bar{y}^\top G_{yu})$, $G_u^\top G_u$ and $\bar{y}^\top G_{uu}$ and the entries of the gradient L_u are not significantly larger or smaller than in the case where $\dim(u) = 6$. In B and L_u, the sixth row/column still contributes the dominating values. Consequently, it is the additional dimension and the influence of the sinking velocity exponent u_7 on the system of model equations that lead to changes in the length of the steps.

We observe the model being very sensitive with respect to parameter u_7. A closer look at parameter values during the optimization process indicates u_7 strongly influencing u_6 which is at its maximum value when u_7 is at its minimum value and vice versa.

Analyzing the model equations shows that u_7 introduces a new composition and correlation of entries of the Jacobian $G_u \in \mathbb{R}^{n\times 7}$ compared to the Jacobian $G_u \in \mathbb{R}^{n\times 6}$ in testings with fixed u_7. The sinking velocity exponent $u_7 = b$ occurs in the coupling equation q_N in the non-euphotic zone. See the appendix B for an overview of the model equations, the spin-up and their derivatives. An incautiously chosen step in u_7 is carried forward to the other parameters and therewith to the state vector.

The One-shot strategy was specified to work without any line search procedure. The length of the steps shall only be influenced by the preconditioner B and its factors defined in (2.9) and (2.11) or (2.12). Since we proposed a simplification in that the contraction factor ρ is fixed to an estimated value (due to a lack of knowledge in most

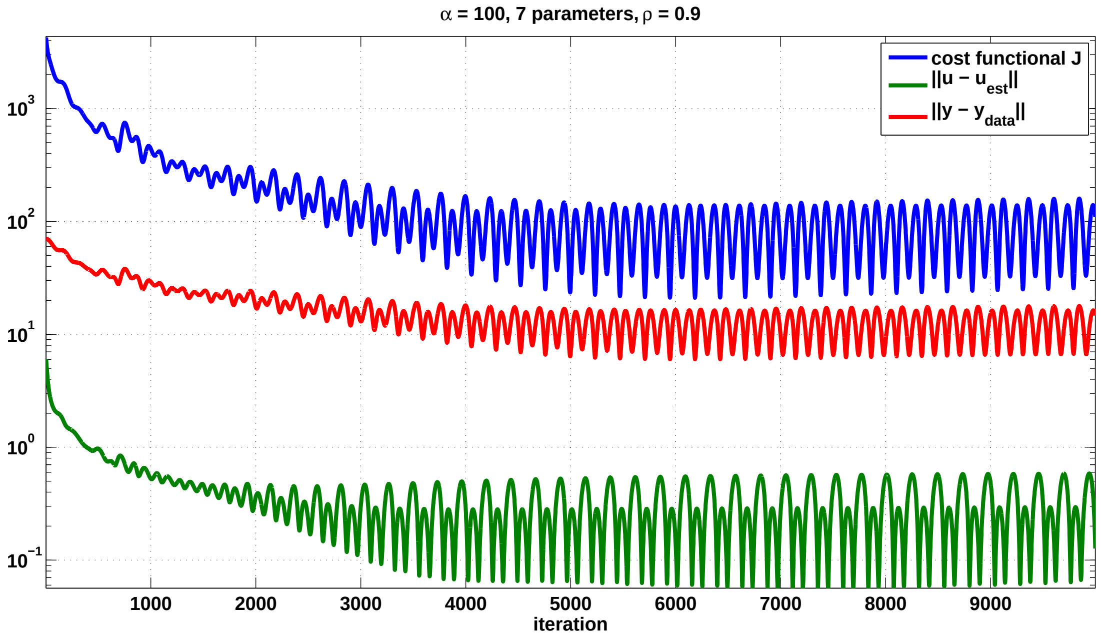

Figure 4.11: Cost functional J, $\|u - u_{est}\|$ and $\|y - y_{data}\|$ during the Oneshot optimization run with 7 parameters. $\alpha = 100$, $u_{est} = u_{opt}$.

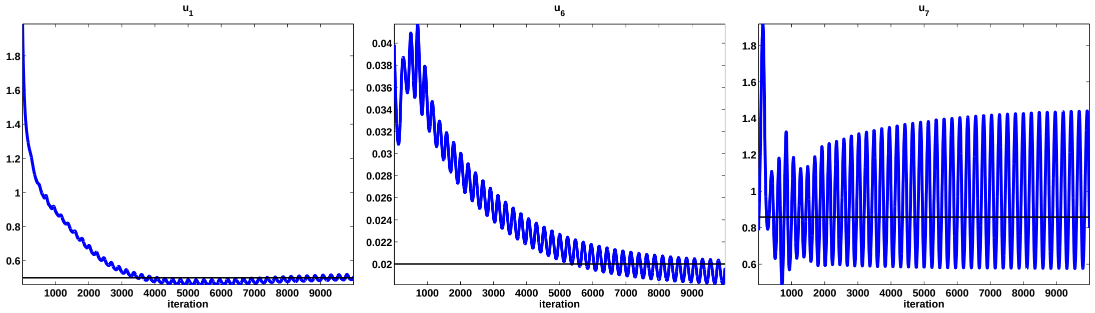

Figure 4.12: Parameter values during the optimization run with 7 parameters, $\alpha = 100$, $u_{est} = u_{opt}$. We exemplarily show u_1, u_6 and u_7 (from left to right).

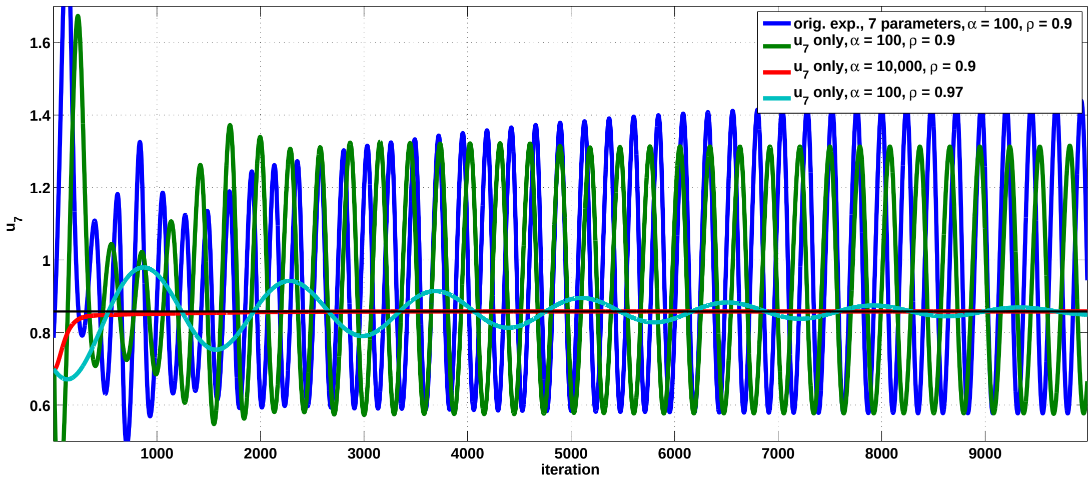

Figure 4.13: Values of the sinking velocity exponent, parameter u_7, during the optimization process for different optimization settings.

Table 4.6: Overview of optimization results for different choices of weighting factors of the preconditioner B. For the latter we compare the behavior of the method with respect to the occurrance of oscillation during the process, convergence of the strategy and quality of the fit.

	$\rho = 0.9$	$\bar{\rho} = 0.998$	$\rho = 0.97$	$\rho = 0.98$	$\rho = 0.99$			
α_L	400	99,999	4,444	9,999	40,000	400	40,000	4,000
β_L	2	2	2	2	2	2	2	2
σ	0.05	0.001	0.015	0.01	0.005	0.005	0.05	0.005
Oscillation	yes	no	yes	damped	no	yes	damped	damped
convergence	no	yes	no	yes	yes	no	yes	yes
fit acceptable	-	no	-	yes	no	-	yes	yes

cases), we suspect that a change in ρ can already solve the problem.

Analytically determining the contraction factor ρ is difficult in most applications. Furthermore, as discussed in chapter 3, usually numerical tests show that even though the contraction factor of the model spin-up exceeds 1 several times, the spin-up still converges to a unique stationary state. For the N-DOP model, we computed the mean value of the contraction factor of an example spin-up:

$$\bar{\rho} := \frac{1}{10,000} \sum_{k=1}^{10,000} \rho_k \approx 0.998 \quad \text{where} \quad \rho_k := \frac{\|y_{k+1} - y_k\|_2}{\|y_k - y_{k-1}\|_2}. \tag{4.5}$$

Using $\bar{\rho}$, computed steps in the parameters are too small such that the One-shot method leads to almost the same results as a model spin-up with the starting values u_0. Hence, we tested other choices of ρ and combinations of α_L, β_L and σ. The corresponding results of the optimization with 7 parameters are collected in table 4.6.

We find that the influential weight in the occurrence of oscillations is σ. In particular, the rate α_L/σ must not be too small. We state that in this example, optimizing seven parameters of the N-DOP model and where the weight α of the cost functional is small compared to the other components of B, α_L and σ must be chosen such that $\frac{\alpha_L}{\sigma}\|G_u\|_2^2$, and therewith also $\|B\|$, is of the order of magnitude of 10^{10} such that the corrections in u_7 are very small, but not too small as in the cases where $\rho \geq 0.99$ and therewith $\frac{\alpha_L}{\sigma}\|G_u\|_2^2 \geq 10^{12}$ and no progress is detected.

A One-shot run with the optimization of u_7 only also leads to an oscillation of the parameter between approximately 0.6 and 1.4 and therefore again no convergence of the strategy. We performed tests with varying α and varying ρ and illustrate the values of u_7 during the optimization in figure 4.13. We can execute a well-performing optimization of u_7 choosing $\alpha = 10,000$, $\rho = 0.9$. The One-shot strategy converges quickly to the optimal $u_7^* = 0.858$ and the state y follows to the optimal state y^* respectively.

This confirms the analysis of section 4.3.1, where we conclude that in this optimization problem with the N-DOP model α has to be chosen very large to gain any influence on the preconditioner B and the gradient L_u and therewith on the course of the optimization. In contrast to tests with $\alpha = 100$, in this case, the dominant part of L_u is J_u and not $\bar{y}^\top G_u$ and we observe L_u converging to zero without oscillation. Furthermore, in this case α adds the leading dimension to B. A test choosing $\alpha = 10,000$ certainly is not meaningful concerning the applicability in problems where optimal parameters are not known. Hence, tuning the parameters of the strategy via ρ is necessary and as illustrated in figure 4.13 it is successful for $\rho = 0.97$ in the optimization of the parameter u_7 only.

As summarized in table 4.6, tests with seven model parameters converge after adjustment of ρ to $1 > \rho \geq 0.98$, but they are not as successful as the optimization of u_7 only. The reduction of the cost function and parameter values are illustrated in figures 4.14 and 4.15. Note that far more iteration steps are necessary to converge and that the obtained cost function value and parameters are not optimal at all. Restarts neither with parameters and state variables from results of the shown experiment using a different adjoint state nor restarts with parameters from results changing state *and* adjoint state do improve the results.

In contrast to the optimization approaches of Prieß et al. [32] and Piwonski [30], we include data of one time point only in the cost functional. Using the equation error approach, Piwonski [30] can reproduce a whole trajectory of synthetic data with 2,880 time points finding optimal parameters. Prieß et al. [32] are able to reduce the cost functional significantly fitting synthetic data of 45 time points applying the surrogate based optimization. The surrogate based optimization does not converge to optimal parameters, but the algorithm has no difficulty finding the optimal parameter u_7^*. In section 4.5, we will investigate the performance of the One-shot strategy including four time-points in the cost functional.

4.3.4 Bounded Retardation and Results on computational time

The main goal of the One-shot iteration is to find optimal parameters with bounded retardation, i.e. after a number of iteration steps which is only a small multiple of the number of iteration steps needed for the model spin-up. We considered a model spin-up to be feasible when for the periodic solution the state of the tracers at one time point changes imperceptibly, i.e. $\|y - y_{data}\| < 10^{-5}$. For a spin-up initialized with $(y_N)_i = 2.17$ and $(y_{DOP})_i = 0.0001$ for $i = 1, ..., \dim(y_N)$, at all considered points in the

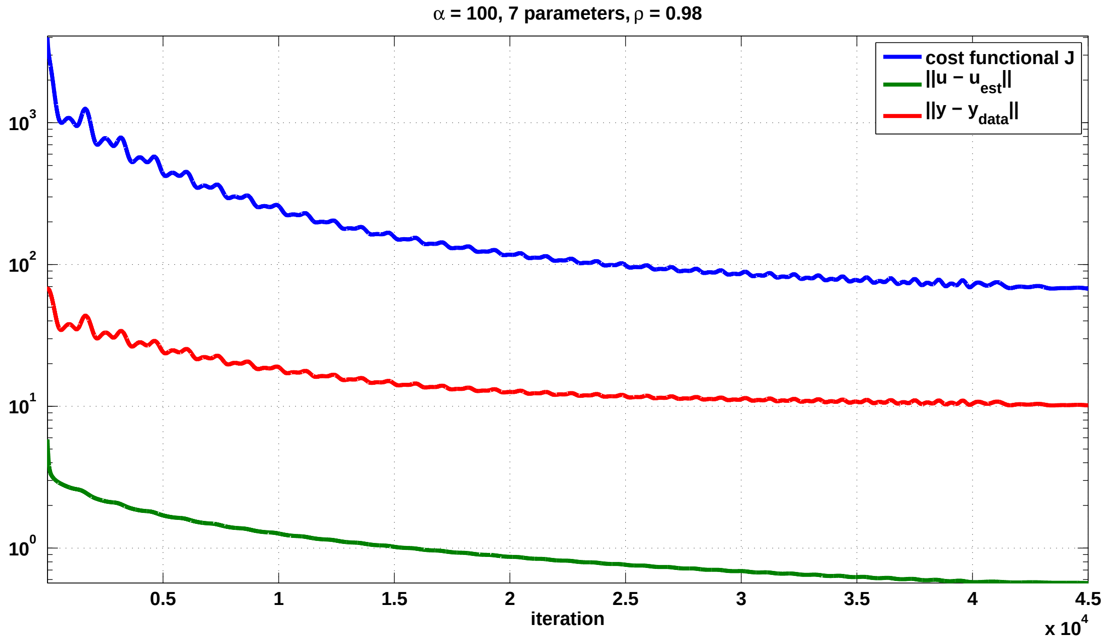

Figure 4.14: Cost functional J, $\|u - u_{est}\|$ and $\|y - y_{data}\|$ after the adjustment of ρ. $\rho = 0.98$, $\alpha = 100$, $u_{est} = u_{opt}$.

ocean a periodic state with a steady annual cycle is reached after 10,000 iteration steps. Starting with the output concentrations of a spin-up with slightly different parameters still demands approximately 5,000 to 7,000 model years.

Results of the twin experiment with 6 parameters showed that acceptable results are obtained after 15,000 or 20,000 One-shot steps, which indeed is only a small multiple of the number of spin-up steps. For 7 parameters, 45,000 steps were needed to converge which still is a factor of only 4.5. Of course not only the number of iterations is relevant but also the comparison of the computational times.

In the following, we compare the computational time for the model spin-up, the model spin-up including the computation of the adjoint state and the One-shot iteration optimizing 6 parameters. We computed on 2 nodes of Intel Sandy-Bridge-processors with 16 cores. Table 4.7 summarizes the results.

For the whole optimization run of the optimization problem from section 4.3.2 with 20,000 One-shot iteration steps, we measured a computational time of 417:12:14 h which is only 13.8 times the time needed for 10,000 spin-up years. This includes the simplification in that the update of the preconditioner B is performed only every 5th step (*inexact One-shot steps*) after 500 steps with full computation of B (*exact One-shot steps*). This reduced the computational time to 9:26:06 h for 500 inexact One-shot

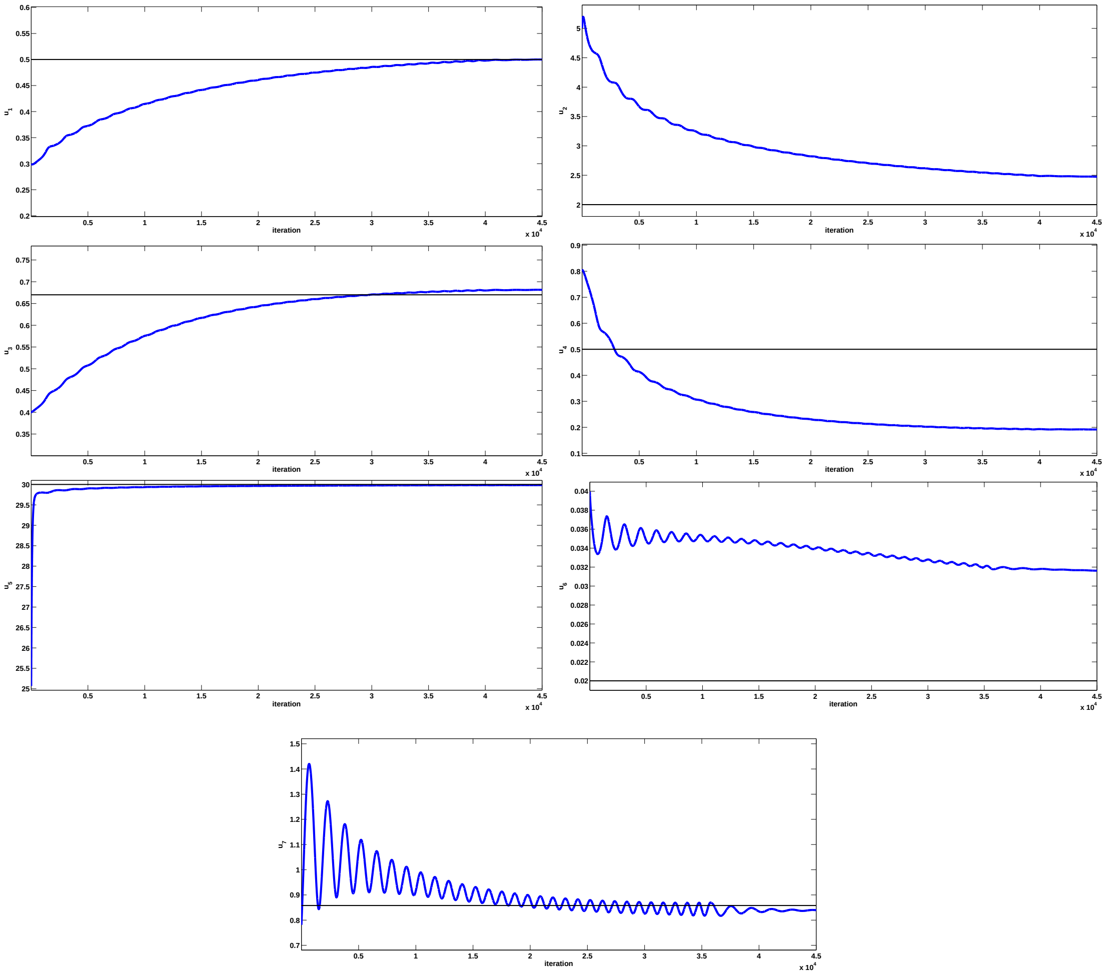

Figure 4.15: Parameter values during the optimization run after the adjustment of ρ. $\rho = 0.98$, $\alpha = 100$, $u_{est} = u_{opt}$.

Table 4.7: Computational time for the spin-up, the spin-up augmented by the update of the adjoint state and the One-shot method with 6 parameters to be optimized. The overall time until acceptance of the One-shot strategy includes 500 Pre-Iterations, 500 exact One-shot steps and inexact One-shot steps until acceptance (here: 20,000 iteration steps). Computation is performed on two Intel Sandy-Bridge processors with 16 cores.

	elapsed time of 1 step	factor	elapsed time of 500 steps	factor	elapsed time until acceptance	factor
Spin-up	10.9 s		1:27:31 h		30:13:07 h	
Spin-up and adjoint state	43.4 s	4.0	5:41:35 h	3.8	113:15:12 h	3.8
Exact One-shot	292.4 s	26.8	39:56:07 h	27.4	417:12:14 h	13.8
Inexact One-shot			9:26:06 h	6.5		

iteration steps and therewith to a factor of 6.5 compared to the model spin-up and of only 1.7 compared to the augmented model spin-up.

Table 4.7 lists results for the optimization of 6 model parameters. A factor of 1.2 has to be multiplied for respective One-shot tests with 7 parameters. A One-shot optimization with 7 parameters and 45,000 iteration steps demands approximately 1,061.3 h which corresponds to the time needed for 35.1 spin-ups.

4.4 Data from the World Ocean Atlas

In this section, we consider data from the World Ocean Atlas 2005 [6] and the World Ocean Database 2013 [3]. Here, only the concentrations of PO4 (or equivalently N) are included in the cost function, because data for DOP is available only at a negligibly small number of points in the ocean. The measuring of DOP is very expensive compared to the measuring of phosphorus PO4.

However, the state y still includes the values for DOP concentrations such that the model spin-up remains the same as in the previous tests. Only the cost functional J and its derivatives have to be adjusted.

In first tests, we stay very close to previous testings in that we include data of one time point only but linearized to all spatial points of the discretization in the cost functional. We use data sets which correspond to monthly means of January, April, July and October. The results are summarized in table 4.8, where we collect results for the optimization with 6 parameters fixing u_7 to $u_7 = 0.858$ and 7 parameters respectively.

We observe that tests perform very similarly for the different dates in that the cost function is reduced by a factor of approximately 1.5 only. The fit is reduced by an even worse factor of 1.2 only. We chose the same strategy parameter $\rho = 0.9$ as in the previous tests with 6 parameters and $\alpha = 1$ to only slightly keep parameters near the estimated values. In tests with 7 parameters we set ρ to $\rho = 0.98$. As tests with 7 parameters perform very similarly for January and July, we omit testings for April and October.

The results after 10,000 One-shot iteration steps are collected in figures 4.16, 4.17 and table 4.8. We terminate after 10,000 iteration steps as the fit does not improve anymore even though parameters are not fully converged. However, a rough estimation of final parameter values is assessable.

For the different data sets, the strategy tends to different parameter sets. Parameters u_1, u_2, u_3, u_6 differ perceptibly in tests with 6 parameters only. The difference of the obtained parameters is even greater compared to tests with 7 parameters. However, the fit is not further improved in tests with 7 parameters. We detect that even though the obtained parameters differ significantly in the testings, the largest differences to data are measured in the same regions for all data sets, namely in the third and fourth ocean layer at the coastal regions of the North Pacific. See figure 4.18 and 4.19 for illustration. Note that compared to tests with synthetic data from section 4.3.2 illustrated in figure 4.10 the scale of the measured differences to given phosphorus concentrations is 10^3 times larger.

4.5 Optimization Problem with Multiple Time Points

Until now, we have analyzed the case where data of one time point, in most cases January 1st, 0:00 a.m., given at all discrete points in the ocean is included in the cost function. The state y is considered at one time point and is driven towards a stationary state (and therewith an annually periodic trajectory) via the model spin-up. In this chapter, we take periodicity and several time points of the trajectory into consideration. Concerning periodic solutions, Günther et al. [15, 2] investigate the One-shot strategy in the optimization of the flow around a cylinder with unsteady Reynolds-averaged Navier-Stokes equations (URANS). In [15, 2], for URANS they have successfully performed the piggy-back iteration for the state and adjoint state variable (for fixed parameters u). For n_d time points, they set up a new fixed point iteration function H with $H : Y^{n_d} \times U \rightarrow Y^{n_d}$ in that for each time point $i = 0, 1, ..., n_d - 1$ a steady state y_i^* is searched for by performing the model spin-up under the additional constraint that for the parameter set $u \in U$ the concatenation of the states y_i^* to a trajectory fulfills the model equations.

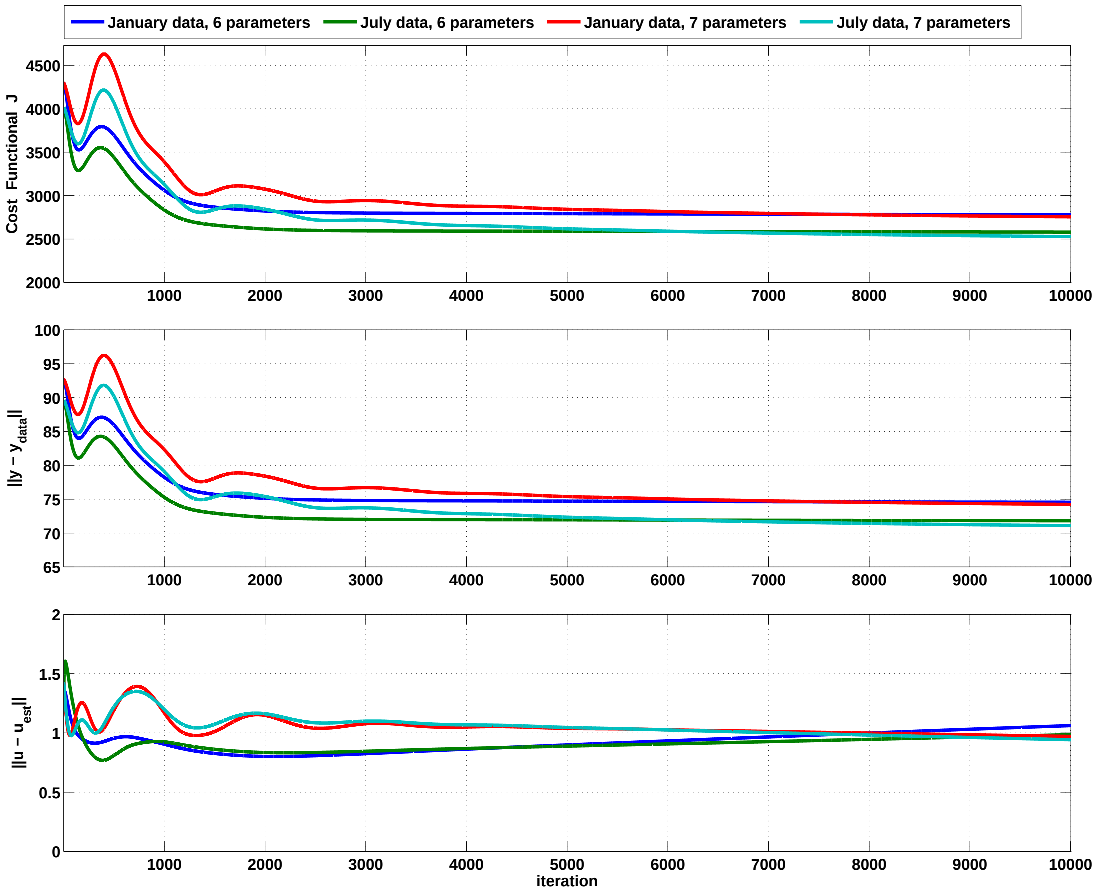

Figure 4.16: Reduction of the cost functional J for fitting WOA data, $\alpha = 1$. In contrast to previous plots, here the y-axis has a linear scale.

Table 4.8: Overview of initial parameter set u_0, the estimated values u_{est}, function values at initial parameter and state variables, computed function value, data fit and parameter values after 10,000 One-shot steps for WOA data, $\alpha = 1$.

	$J(y_0,u_0)$	$J(y^*,u^*)$	fit	u_1	u_2	u_3	u_4	u_5	u_6	u_7
u_0				0.4	3.0	0.65	0.5	31.0	0.03	0.858
u_{est}				0.5	2.0	0.67	0.5	30.0	0.02	0.858
January	4,295.4	2,779.4	74.55	0.9149	2.561	0.704	0.0560	30.66	0.02289	
April	4,176.4	2,716.4	73.70	0.7260	2.195	0.658	0.0444	30.74	0.02327	
July	4,017.7	2,579.5	71.82	0.8016	2.368	0.676	0.0431	30.73	0.02393	
October	4,184.9	2,731.8	73.91	0.8163	2.443	0.679	0.0555	30.70	0.02389	
January	4,295.4	2,755.6	74.23	0.3980	2.833	0.515	0.432	30.45	0.02532	0.9368
July	4,017.7	2,527.9	71.10	0.3858	2.753	0.504	0.415	30.51	0.02510	0.9521

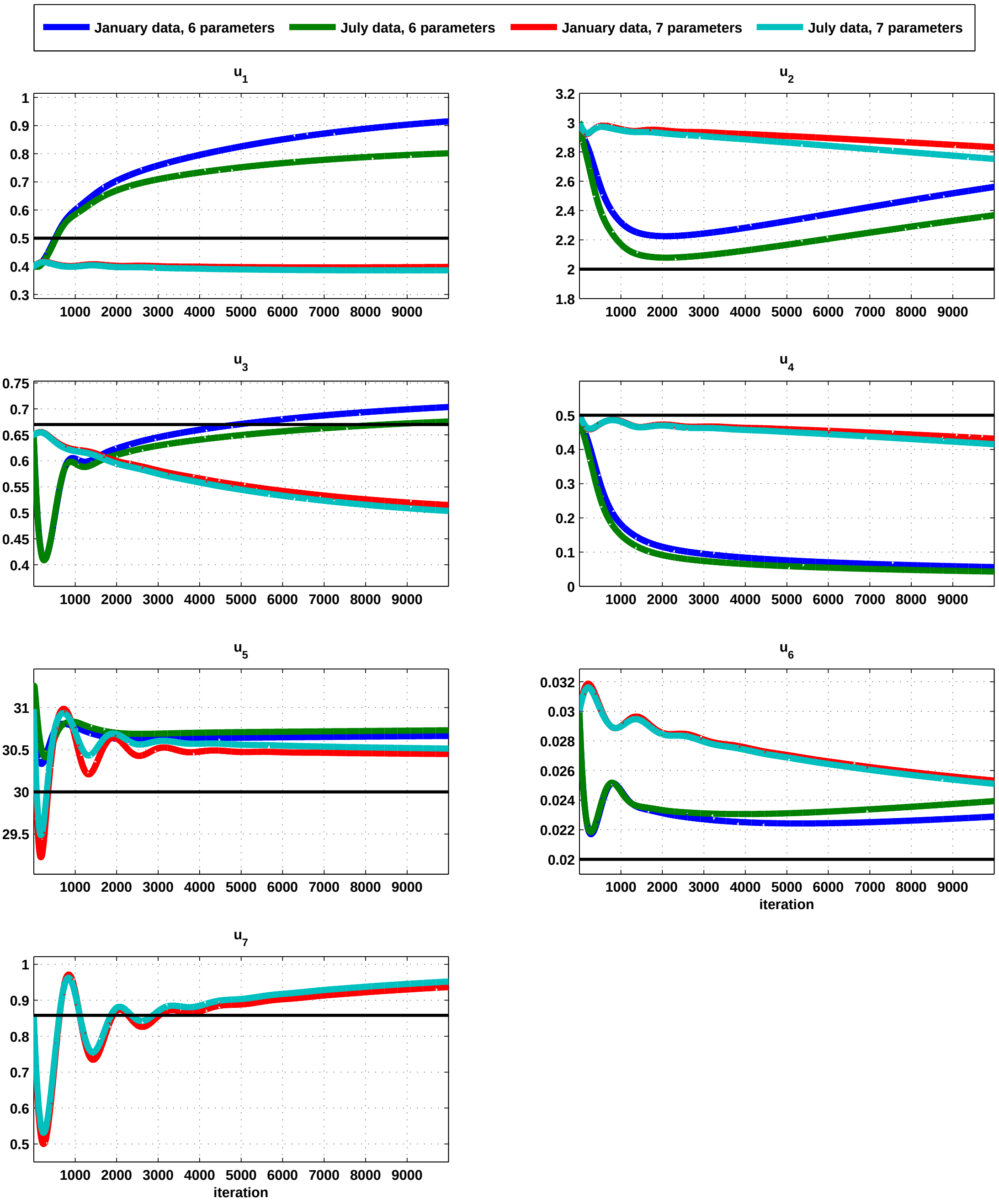

Figure 4.17: Parameter values during the optimization run fitting World Ocean Atlas Data, $\alpha = 1$.

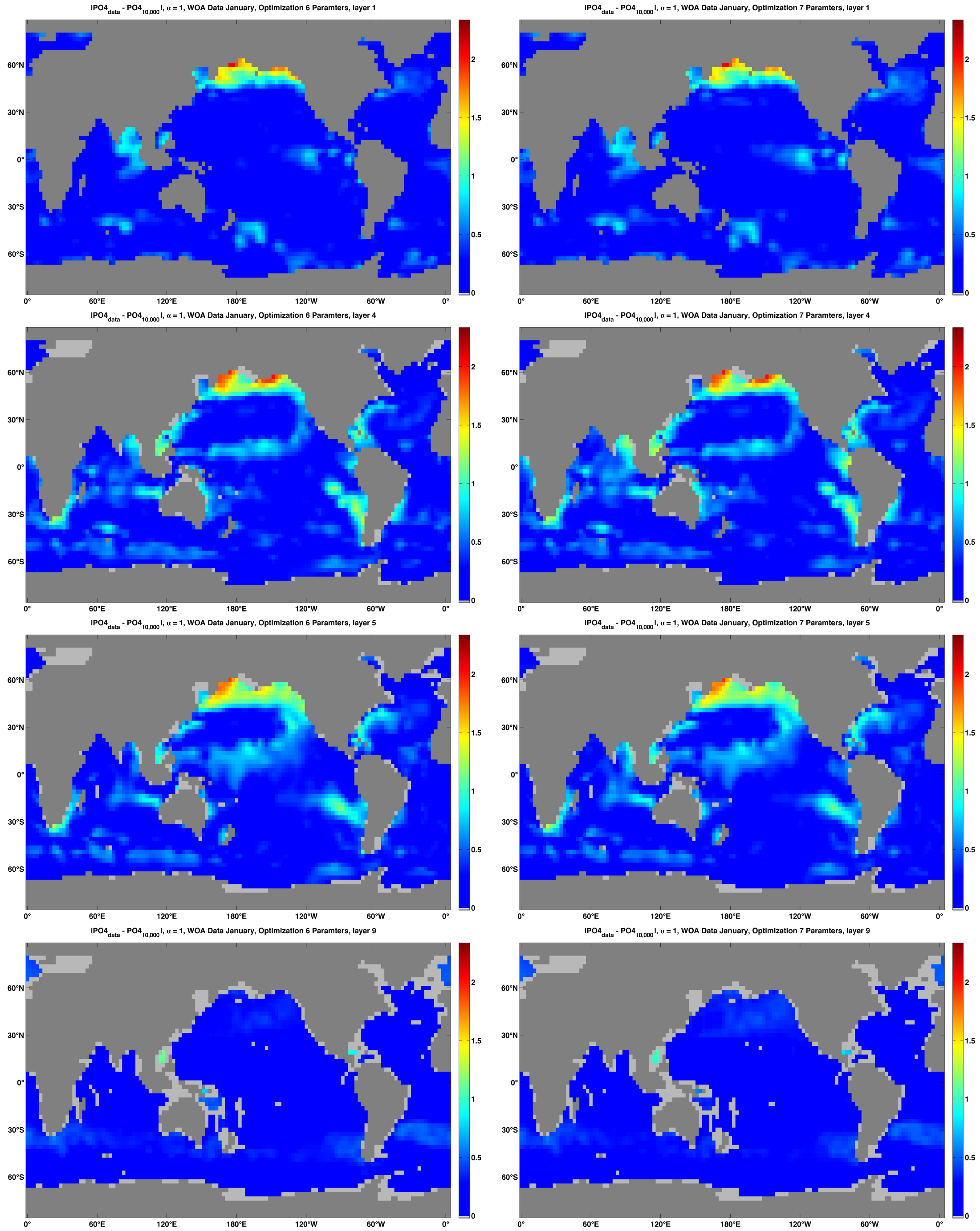
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 6 Paramters, layer 1
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 7 Paramters, layer 1
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 6 Paramters, layer 4
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 7 Paramters, layer 4
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 6 Paramters, layer 5
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 7 Paramters, layer 5
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 6 Paramters, layer 9
|PO4$_{data}$ - PO4$_{10,000}$|, α = 1, WOA Data January, Optimization 7 Paramters, layer 9
60°N
30°N
0°
30°S
60°S
0°
60°E
120°E
180°E
120°W
60°W
0°
2
1.5
1
0.5
0

Figure 4.18: Misfit of the concentrations of PO4 (equivalent to N) compared to data (January) from the World Ocean Atlas 2005 after 10,000 One-shot iteration steps. Left: Results of the optimization including 6 model parameters. Right: Results of the optimization including 7 model parameters. Plots are shown exemplarily from top to bottom in the uppermost layer (0-50 m b.s.l.), the fourth (220-360 m b.s.l.), the fifth (360-550 m b.s.l.) and ninth layer (1420-1810 m b.s.l.).

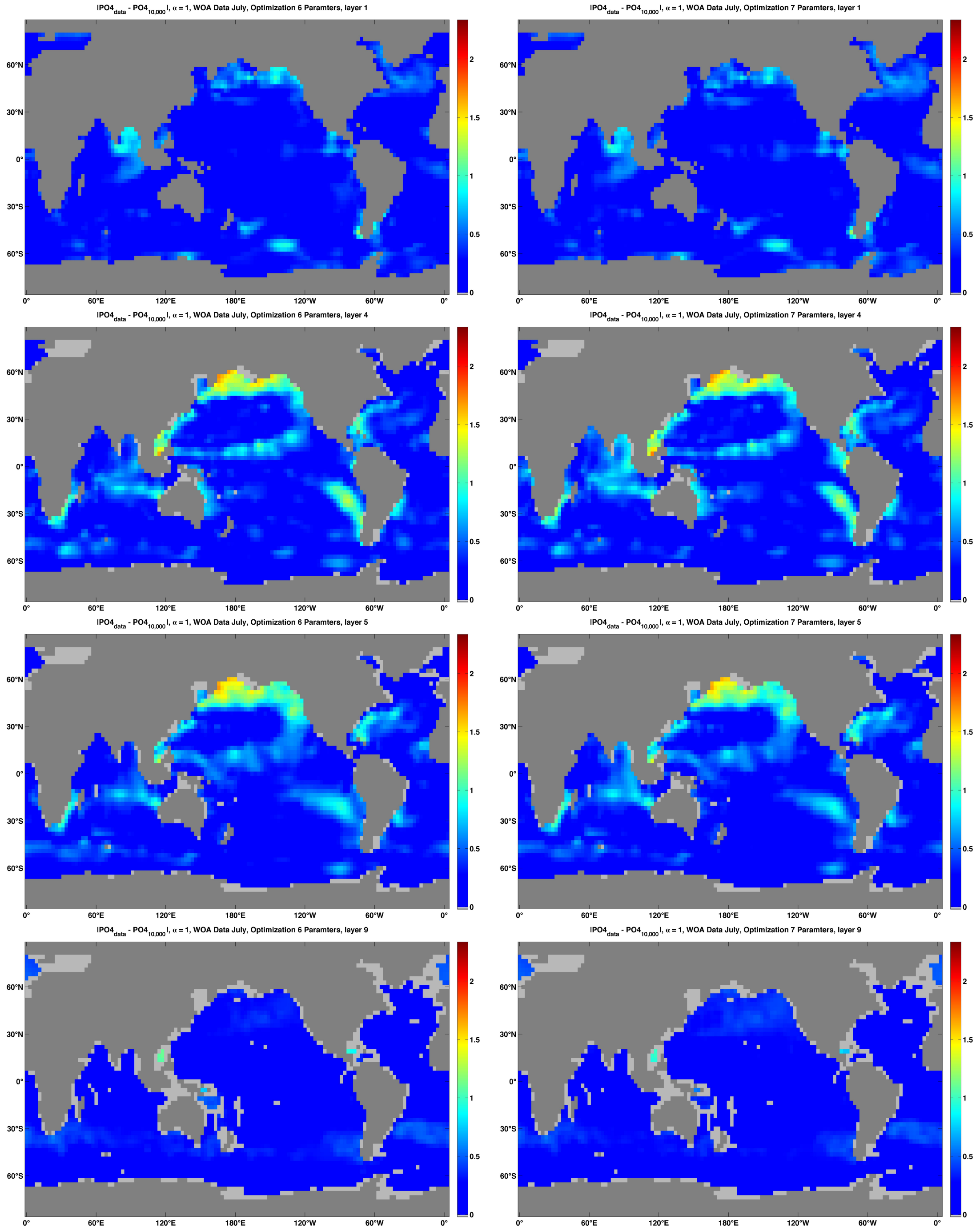
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 6 Paramters, layer 1
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 7 Paramters, layer 1
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 6 Paramters, layer 4
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 7 Paramters, layer 4
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 6 Paramters, layer 5
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 7 Paramters, layer 5
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 6 Paramters, layer 9
$|PO4_{data} - PO4_{10,000}|$, $\alpha = 1$, WOA Data July, Optimization 7 Paramters, layer 9
60°N
30°N
0°
30°S
60°S
0°
60°E
120°E
180°E
120°W
60°W
0°
2
1.5
1
0.5
0

Figure 4.19: Misfit of the concentrations of PO4 (equivalent to N) compared to data (July) from the World Ocean Atlas 2005 after 10,000 One-shot iteration steps. Left: Results of the optimization including 6 model parameters. Right: Results of the optimization including 7 model parameters. Plots are shown exemplarily from top to bottom in the uppermost layer (0-50 m b.s.l.), the fourth (220-360 m b.s.l.), the fifth (360-550 m b.s.l.) and ninth layer (1420-1810 m b.s.l.).

They formulate the resulting One-shot procedure as the original One-shot strategy (2.7) replacing G with the new fixed point iteration H. Furthermore, the authors investigate an adaptive time scaling approach and, applying this, they successfully optimize an inverse design problem subject to the Van der Pol oscillator applying the One-shot strategy.

In our case of the N-DOP model, we chose a different approach and propose an idea with which we can include data of arbitrary time points and spacial points. This approach also presents an alternative to the equation error approach applied by Piwonski in [30], where the author constitutes that for the equation error optimization for each component of the state variable or vector y corresponding data has to be available. As this is not the case for N and DOP, the approach is not applicable straight away with real world data.

So far, the considered problem has been to minimize

$$J(y,u) = \frac{1}{2}\|y - y_{data}\|^2 + \frac{\alpha}{2}\|u - u_{est}\|^2 \tag{4.6}$$

such that $G(y,u) = y$ where G is a fixed point iteration function.
The dimension of $y = (y_N, y_{DOP})$ is $\dim(y) = 105{,}498$, which corresponds to data given at one time point in the whole spacial domain. Optimization runs with synthetic data at January 1st, April 1st, July 1st and October 1st, 0:00 a.m. each, were successful in that parameter and state vectors were reproduced. Collecting n_d time points in one state vector y^{n_d} as done in [15, 2] would result in a huge dimension $\dim(y^{n_d}) = n_d * 105{,}498$ and $G : \mathbb{R}^{n_d * 105{,}498} \times \mathbb{R}^{\dim(u)} \to \mathbb{R}^{n_d * 105{,}498}$ which demands a huge amount of storage. We avoid this by taking given points (at given time and given place) into account in the cost function, but keeping the constraint as a function of $y = y_{T_0}$ meaning all points in the ocean at time T_0. This is a reasonable approach because we know about the periodicity of the annual trajectory for a steady state at one time point. We reformulate the problem as follows:
Let $(t_i)_1^{n_d}$ be the set of time points at which data is given and let the vectors $x_i = (x_{i,1}, \ldots, x_{i,\dim(x_i)})$, $i = 1, \ldots, n_d$, be the collection of the spacial indices of the tracer concentrations where at time point t_i data is given. Let $\mathcal{D} = \{y_{(d,x_1,t_1)}, \ldots, y_{(d,x_{n_d},t_{n_d})}\}$ be the set of given data, i.e. $\dim(y_{(d,x_i,t_i)}) = \dim(x_i)$, corresponding to the number of

data points given at time t_i. The problem then is to minimize the function $J(y,u)$

$$J(y,u) = \frac{1}{2} \left\| \begin{pmatrix} G_{(x_1,t_1)}(y,u) - y_{(d,x_1,t_1)} \\ G_{(x_2,t_2)}(y,u) - y_{(d,x_2,t_2)} \\ \vdots \\ G_{(x_{n_d},t_{n_d})}(y,u) - y_{(d,x_{n_d},t_{n_d})} \end{pmatrix} \right\|^2 + \frac{\alpha}{2}\|u - u_{est}\|^2, \tag{4.7}$$

such that $G(y,u) = y$ where G remains the spin-up function with $n_t = 2{,}880$ intermediate time steps as we have described in detail in the appendix B with $G : \mathbb{R}^{105{,}498} \times \mathbb{R}^{\dim(u)} \to \mathbb{R}^{105{,}498}$. So on the one hand, in iteration step k the state vector y_k again collects the tracer concentrations of N and DOP at one arbitrary time point T_0 of the year at all 52,749 points in the world ocean. On the other hand, the functions $G_{(x_i,t_i)}(y_k,u_k)$ compute the tracer concentration of N and DOP respectively at time t_i depending on the tracer concentrations at the chosen starting time point T_0 and select the spacial points x_i:

$$\begin{aligned} G_{(x_i,t_i)} : \mathbb{R}^{\dim(y)} &\times \mathbb{R}^{\dim(u)} \to \mathbb{R}^{\dim(x_i)} && (4.8) \\ G_{(x_i,t_i)}(y_k,u_k) &= (y_{k,t_i})_{x_i} \text{ where} && (4.9) \\ y_{k,j+1} &= A_{imp,j}(A_{exp,j}y_{k,j} + q_j(y_{k,j},u_k)), \text{ for } j = 0, ..., t_i - 1. && (4.10) \end{aligned}$$

In the One-shot optimization context, the algorithm is different only in the computation of the cost functional and its derivatives.

$$J(y,u) = \frac{1}{2}\sum_{i=1}^{n_d} \left\| G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)} \right\|^2 + \frac{\alpha}{2}\|u - u_{est}\|^2 \tag{4.11}$$

$$\begin{aligned} J_y(y,u) &= \sum_{i=1}^{n_d} \left(G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)} \right)^\top \frac{\partial}{\partial y} G_{(x_i,t_i)}(y,u) && (4.12) \\ J_u(y,u) &= \sum_{i=1}^{n_d} \left(G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)} \right)^\top \frac{\partial}{\partial u} G_{(x_i,t_i)}(y,u) + \alpha(u - u_{est}). && (4.13) \end{aligned}$$

At first glance, the vector-Jacobian products again seem to demand the reverse mode of Automatic Differentiation. However, we cannot re-use the intermediate values of $\bar{y}^\top G_y(y,u)$ or $\bar{y}^\top G_u(y,u)$ which we need for the One-shot update, because the reverse computation starts from a different time point with a different input vector multiplied from the left to the Jacobian. That means for the computation of n_d products

$\left(G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)}\right)^\top \frac{\partial}{\partial y} G_{(x_i,t_i)}(y,u)$ and $\left(G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)}\right)^\top \frac{\partial}{\partial u} G_{(x_i,t_i)}(y,u)$ we can use the intermediate values of $G(y,u)$ stored in the forward sweep, but we have to execute n_d reverse sweeps starting from the intermediate time steps t_i, $i = 1, ..., n_d$.

Another theoretical possibility is to carry forward the full Jacobian G_y in time as described in the appendix B and, at time step t_i, multiply the vector $\left(G_{(x_i,t_i)}(y,u) - y_{(d,x_i,t_i)}\right)$ with the selected Jacobian $\frac{\partial}{\partial y} G_{(x_i,t_i)}(y,u)$ and $\frac{\partial}{\partial u} G_{(x_i,t_i)}(y,u)$ respectively. But setting up the full Jacobian carrying forward in time is too expensive with respect to storage amount.
If there is only a small number of time points for which a certain number of data points is given, calling the reverse mode n_d times proves to be the best solution.

In general, data is not given equally distributed in time. For N and DOP needed in this model, there is significantly more data in July (57,995 phosphate values according to the World Ocean Data Base 2013, [3]) than in January (40,469) and December (29,352). Thus, in the context of the considered N-DOP model, it is advantageous to first of all analyze given data and select a time point T_0 from which the model spin-up is started (with the corresponding transport matrices and coupling functions) requiring the fewest time integration steps to reach the time points t_i, $i = 1, ..., n_d$ where data is available.
Still, there is the need for computing the second order derivatives J_{yu} and J_{uu} for the computation of the preconditioner B. As we perform the finite difference (FD) approach over $\bar{y}^\top G_y$ and $\bar{y}^\top G_u$ obtaining $\bar{y}^\top G_{yu}$ and $\bar{y}^\top G_{uu}$ respectively, we augment the FD call to a call over $J_y + \bar{y}^\top G_y$ and $J_u + \bar{y}^\top G_u$ to obtain the required second order derivatives.

4.5.1 Numerical Results considering Four Time Points

In our testing scenario, we include data of phosphorus N at 4 time points, January 1st, April 1st, July 1st, October 1st, each at 0:00 a.m, at all points of the world ocean.
Reproducing synthetic data computed with optimal values u_{opt} from table 4.1 at the four time points performs very well and converges even faster than in comparable tests (choosing the same weight $\alpha = 100$, and the same $\rho = 0.9$, optimizing 6 parameters) with one time point only. After 10,000 One-shot iteration steps, results with multiple time points are comparably good as those obtained after 16,000 One-shot steps with data of one date only. This is not surprising as the given synthetic data (matching the same optimal parameter set) provides more information on the system and the annual cycle if more time points are included. Figure 4.20 illustrates the results. Due

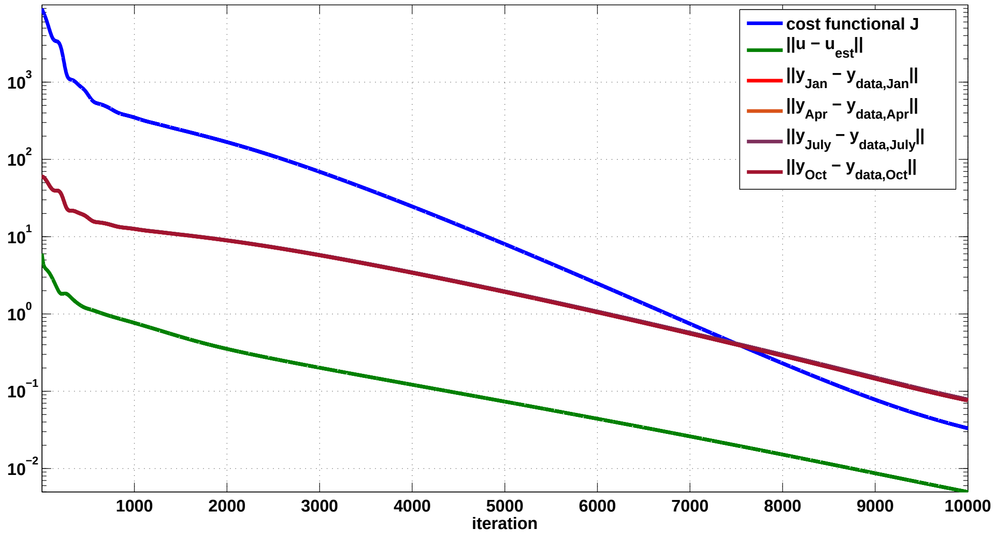

Figure 4.20: Cost functional J and its parts using synthetic data at four time points, $\alpha = 100$, $u_{est} = u_{opt}$.

to choosing synthetic data, the fits at the four time points are almost equal.

Inclusion of the sinking velocity exponent u_7 in the parameter vector to be optimized, results in the same oscillating output as in section 4.3.3.

Considering computational time, we measure that the time needed for one exact One-shot step enlarges the time by a factor of 2.3 in tests with 6 parameters and by a factor of 2.4 in tests with 7 parameters. For five inexact One-shot steps, we measured computational times enlarged by factors of 2.26 for 6 parameters and 2.6 for 7 parameters comparing to previous tests with one time point only. However, as mentioned above, in tests with four time points and synthetic data, we only need 0.625 times the number of One-shot steps to obtain the same quality of the fit. Furthermore, we obtain the results in only one optimization run.

In the following, we consider real world data given as monthly means of January, April, July and October from the World Ocean Atlas 2005 [6]. We observe that compared to tests with one time point and 6 parameters from section 4.4, the fits do not improve (see figure 4.21, and table 4.9). The obtained parameter set again is different from those obtained in tests with single time points at the respective dates. See figure 4.22 for illustration of the parameter values during the optimization run. Interestingly, in this test, the parameter values pass through larger intervals of computed values during the optimization process than in previous testings with one time point only. Espe-

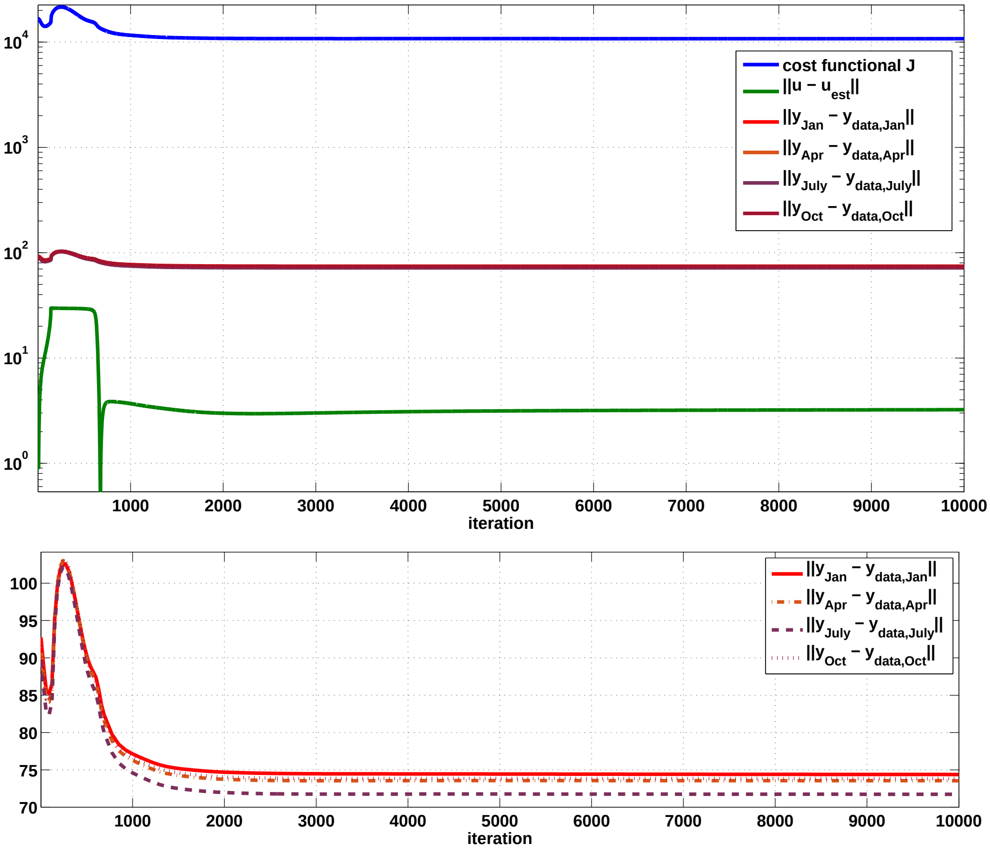

Figure 4.21: Top: Cost functional J and its parts using WOA data at four time points, January, April, July, October. Bottom: Norm of the differences to data at the four different time points during the optimization process.

cially parameter u_5 almost hits zero even though in all other testings with WOA data and weighting factor $\alpha = 1$, the value has never shown this behaviour. However, the strategy returns to values near the estimated optimal value without changing any parameters of the optimization algorithm as for example the weight α or the contraction factor ρ.

Just as in tests with synthetic data, in this case with WOA data the One-shot strategy again computed the same fits as in four corresponding individual optimization runs with one time point only. That means even though the computational time for one One-shot step with four time points is more than 2.26 times longer than a One-shot step with one time point, the user saves computational time as only one optimization run is to be performed.

Table 4.9: Overview of initial parameter set u_0, the estimated values u_{est}, function values at initial parameter and state variables, computed function value, data fit and parameter values after 10,000 One-shot steps for WOA data, $\alpha = 1$.

	$J(y_0,u_0)$	$J(y^*,u^*)$	fit (Jan/Apr/July/Oct)	u_1	u_2	u_3	u_4	u_5	u_6
u_0				0.4	3.0	0.65	0.5	31.0	0.03
u_{est}				0.5	2.0	0.67	0.5	30.0	0.02
	16,694.7	10,775.1	74.4 / 73.8 / 71.7 / 73.8	0.5885	2.578	0.662	0.0128	33.13	0.0269

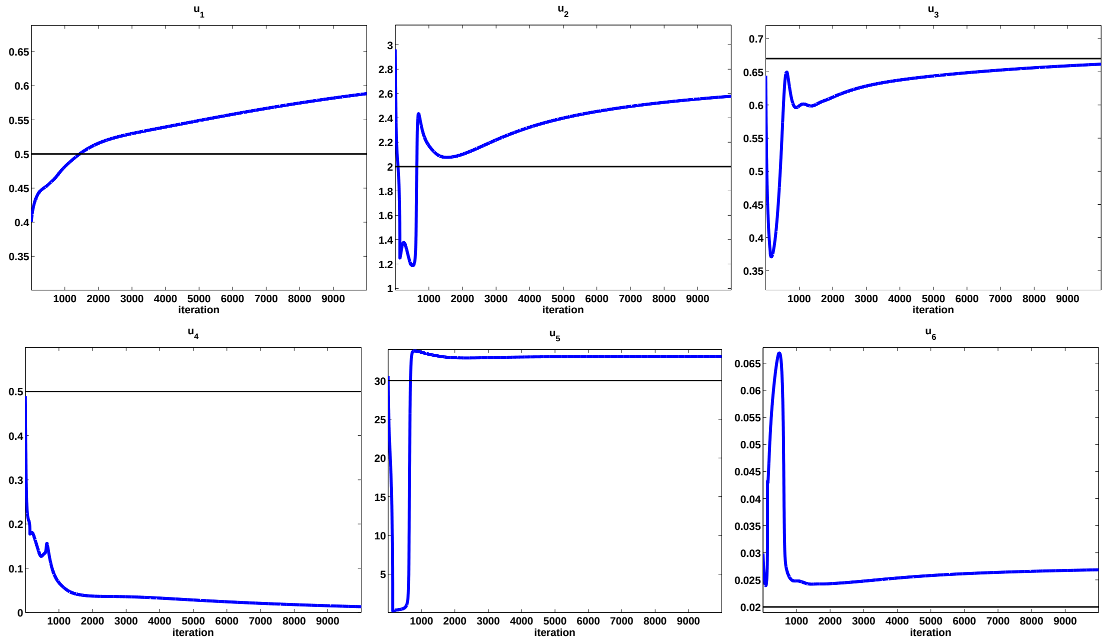

Figure 4.22: Parameter values during the optimization process for the optimization of 6 parameters and with four data sets given in January, April, July and October. Here $\alpha = 1$, $\rho = 0.9$.

5 Summary and Conclusions

In this chapter, we summarize and emphasize main results concerning the assessment of the One-shot optimization strategy in the calibration of marine ecosystem models. From our experiments with the One-shot strategy, we will not provide suggestions on possible adjustments of parameters of the specific models. We neither consider questions arising in marine geosciences on projections of the future ocean the optimized models may provide.
We rather deal with questions concerning the applicability of the One-shot strategy on models used in marine sciences.

The major result is: The One-shot optimization strategy is applicable (though with adjustments and simplifications) in the calibration of marine ecosystem models. For models requiring an expensive and extensive spin-up, the One-shot method proves to be an effective strategy in the optimization of model parameters as it corrects parameters in hand with the model spin-up.

We emphasize that the One-shot strategy does not intervene in the iterative process to solve the model equations. The fixed point iteration $G : Y \times U \to Y$ to compute a feasible state y^* is assumed to be given and already found reliable for fixed model parameters.
In this work, we have analyzed the performance of the One-shot strategy on the Rahmstorf 4-box model as a representative of a low-complexity model with an explicit Euler time stepping and the N-DOP model representing a model of intermediate complexity. Here, on the one hand, complexity means the complexity in the parametrizations in the different models. The complexity is increased by adding more state variables and detailed parametrizations of additionally considered processes. On the other hand, complexity means the discretization of the temporal and spatial scaling.

We find that even in the case of the low-complexity model, the One-shot method requires simplifications to be applicable. The necessity of simplifications is due to unknown, expensively computable or theoretically not suitable parameters of the One-shot method. In particular, the contraction factor ρ, which is assumed $\rho < 1$, exceeds 1 for several time integration steps during the entire model spin-up. We proposed to fix it to a number close to but less than 1. In the optimization of parameters of the box model, setting $\rho = 0.9$ was adequate. However, for the N-DOP model, the contraction factor ρ which there too exceeds 1 during the time integration, had to be chosen more carefully as we fixed $\|L_{yy}\| = 1$ and the closely related weighting factors σ, α_L and β_L in the computation of the preconditioner B aswell. These factors determine the convergence of the strategy similar to a line search procedure and possibly need to be adjusted during the optimization process. The weights can be corrected during the optimization run without restarting the whole optimization run. Still, the introduction of an appropriate line-search procedure would be advantageous for future testings in cases, where the contraction factor is not known or exceeds 1 for several iteration steps. For future convergence analysis of the One-shot strategy, we propose to consider weaker assumption on the spin-up function G as in both test cases, the One-shot strategy converges, even though the main assumption $\|G_y(y, u)\| \leq \rho < 1$, for all $y \in Y$ and fixed $u \in U$, is not satisfied.

We introduced so-called pre-iterations to stabilize the performance of the optimization. Since, in the beginning, the iterative solver G quite likely performs irregular updates in the state y which possibly are corrected during the successive iteration steps, we only update the state y and the adjoint state $\bar{y}$ and leave the parameters fixed until the iteration starts to exhibit a regular update pattern. This prevents the One-shot strategy from inadequately adjusting parameters in the beginning.

In the considerations of the low-complexity box model, we did not only test the One-shot version with full computation of the preconditioner B but also a version in which B is approximated by a BFGS update applied on the augmented Lagrangian L^a. We find that the One-shot-BFGS approach is not as reliable as the version with full computation of B. It required more iteration steps and therewith more computing time. Furthermore, it required a line-search, especially in the beginning of the iteration process which additionally prolonged the optimization. Hence, we rather concentrated on the original One-shot version in the optimization of the N-DOP model.

We state that the One-shot method does not require any box-constraints. The penalty term $\frac{\alpha}{2}\|u - u_{est}\|_U^2$ of the cost functional replaces additional constraints on the parameters. Depending on the order of magnitude of the components of L_u and B the weight α influenced the optimization process more or less. In the optimization of the 4-box model a weight $\alpha > 0$ was essential whereas in the calibration of the N-DOP model a factor $\alpha = 0$ was permitted.

The nonnecessity of constraints on the parameters to be optimized is ascribed to the following major advantage of the One-shot strategy: It corrects parameter values very cautiously during the optimization run. Improper adjustments often are corrected in the course of the optimization. Contrary to the One-shot method, conventional optimization strategies might compute intermediate parameter sets u_k for which a model spin-up fails and the optimization process terminates with error.

The Karush-Kuhn-Tucker (KKT) optimality condition $\nabla L(y^*, \bar{y}^*, u^*) = 0$ (or in numerical testings $\|\nabla L(y^*, \bar{y}^*, u^*)\| < \varepsilon_{tol}$) was not fulfilled at high accuracy in our testings. $\|L_u(y^*, \bar{y}^*, u^*)\|$ was comparatively large. However, it was reduced significantly. The dominant part of L_u was $\bar{y}^T G_u$ and we observed $\bar{y}$ not being converged at the same accuracy as y. Since the adjoint $\bar{y}$ may converge significantly delayed (up to almost double the number of steps the primal state needs, see for example [2]), we terminate the optimization process when there is no change in y and u detectable.

For both models considered in this work, it was not possible determining *the* optimal parameter set with which data is fit best, except for the twin experiments with synthetic data. On the one hand, the computed parameter sets heavily depended on the chosen cost functional and the penalty term $\frac{\alpha}{2}\|u - u_{est}\|^2$. On the other hand, in particular in the optimization of parameters of the N-DOP model, different parameter sets were computed for different data sets, for a different number of data sets or for a different number of parameters considered in the optimization. However, we have collected a large amount of results, from which possible adjustments of the parameters might be considered. Exemplarily, in the optimization of six parameters of the N-DOP model, we observe the half saturation constant of N, $u_4 = K_N$, moving far away from the estimated value $u_{est,4} = 0.5$ tending to values $u_4^* \approx 0.05$ and even smaller. In addition, we have observed a strong dependency of the system on the sinking velocity exponent u_7. A weighting of the different parameter components could be considered in future testings.

The implementation of the One-shot method requires a lot of first and second order derivative information. These derivatives only depend on the current iterates $(y_k, \bar{y}_k, u_k)$ and not on the entire time integration. We proposed to apply an automatic differentiation (AD) tool, especially for the needed vector-Jacobian products as they can be computed in one call of the reverse mode and without any truncation error. We provided instructions for the computation of derivatives, compared AD derivatives with derivatives computed with the finite differences strategy. We made suggestions for the implementation and for possible inclusion in existing code.

We find that in the low-complexity model of the North Atlantic, the One-shot method was not significantly faster than the conventional BFGS quasi-Newton strategy. It reduced the number of needed Euler steps, but the augmentation to a One-shot step with the computation of the adjoint state $\bar{y}$ and a new parameter set u extends the computational time significantly such that the conventional optimization approach with numerous full (but fast and cheap) Explicit Euler spin-ups could keep up with the One-shot method concerning the computational time.

In the testings with the N-DOP model, we proposed additional simplifications in that the preconditioner B is computed only every fifth iteration step when a certain regular convergence behavior is to be noticed. As we have not compared our results for the N-DOP model to a conventional optimization strategy, we converted the elapsed computing time into the number of full model spin-ups with fixed parameters. We detect, that the elapsed time of 20,000 One-shot iteration steps optimizing 6 model parameters corresponds to only 13.8 full spin-ups with fixed parameters. We assume there is no conventional optimization strategy performing a full model spin-up for each function evaluation, which manages to converge within 13.8 model spin-ups only (using equal initialization). We required very strict accuracy of the fixed point y^* considering the state being converged only after 10,000 model years when the norm of the change in the state was less than 10^{-5}. In Kriest et al. [23] only 3,000 years are considered being sufficient. Using this specification with less model years, still only 46 full model spin-ups are equivalent to 20,000 One-shot steps which then have computed steady states at higher accuracy.

In this work, we presented an own approach to handle periodic systems of equations. For the N-DOP model, the length of the period is known and due to the fixed linear mapping of the model spin-up, finding an annually periodic solution is equivalent to finding a steady state at one arbitrary time point of the period. For consideration of data at intermediate time points we suggested to leave the spin-up function and the corresponding derivatives of G untouched and to include the intermediate time points in the cost functional. Therewith only the computation of the cost function and its derivatives had to be adjusted.

The results were very good in that with one optimization run, we optimized the fit at four different time points as good as four One-shot runs with data of one time-point each. Consequently, the computing time of the One-shot steps was prolonged, but compared to four separately executed optimization runs, it was reduced significantly.

The numerical tests were performed on different platforms of which the fastest avail-

able computed on 128 Intel Xeon E5-4640 processors running at 2.4 GHz. Still an optimization run of 10,000 One-shot steps needed seven days. Future investigations on the One-shot strategy should analyze the performance in test cases which include data at even more intermediate time points. It is almost impossible performing these tests on conventional parallel computing platforms. The augmentation of computing time related to the inclusion of more data at more time points and the quality of the obtained results needs to be investigated.
As we have proposed an approach aiming to fit data at arbitrary temporal and spatial points, future work should include numerical testings with arbitrary spatial points at which data is given. Until now, we have used data linearized to all spatial points of the discretization.

In conclusion, we cannot provide a “manual” for the application of the One-shot method which is universally valid for marine ecosystem models. However, we can provide general recommendations for the implementation and we can realize simplifications making the One-shot method more efficient. Even though the strategy did not perform optimally in all test cases, we have presented tuning possibilities by adjusting criteria of the One-shot strategy itself to make it perform better. In the calibration of marine ecosystem models, the One-shot strategy proves to be a helpful tool and a promising alternative to conventional optimization strategies.

A Publications Related to the Calibration of the 4-Box Model

In the following, we append the publications, which we refer to during the course of this work, related to the application of the One-shot method on the Rahmstorf 4-box model. These are

- the article "Simultaneous model spin-up and parameter identification with the one-shot method in a climate model example" published in *An International Journal of Optimization and Control: Theories & Applications (IJOCTA) 3*, (2013), [22]
- the preprint SPP1253-11-03 "One-shot Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation" published on the preprint server of the DFG-priority program SPP1253 "Optimization with partial differential equations", (2008), [20]
- and the preprint SPP1253-082 "Oneshot Parameter Identification - Simultaneous Model Spin-up and Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation", (2009), [21]

We remark, that in [20] we claim that the explicit Euler scheme for the advanced box model fulfills the contraction property. In the appendix of [20], we intend to show $\|G_y\| < 1$ choosing the $\| \cdot \|_\infty$-norm. We analyse the first line in detail, defining a condition for which the row sum is less than one, and then we wrongly conclude that similar conditions can be derived for the other lines. This is not true. We cannot find appropriate conditions for the fourth and eighth line of G_y. However, the preprint summarizes our first observations in the application of the One-shot method on the Rahmstorf 4-box-model which are worth being attached in this work aswell.

Simultaneous Model Spin-up and Parameter Identification with the One-shot Method in a Climate Model Example

Claudia Kratzenstein[a], Thomas Slawig[b]

[a] Institute for Computer Science, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany
[b] Institute for Computer Science and Kiel Marine Science Centre for Interdisciplinary Marine Science, Cluster The Future Ocean, Christian-Albrechts Universität zu Kiel, 24098 Kiel, Germany

(Submitted to An International Journal of Optimization and Control: Theories & Applications (IJOCTA) October 09, 2012; in final form May 29, 2013; published in IJOCTA Vol.3, No.2; available at: http://ijocta.balikesir.edu.tr/index.php/files/article/view/144/78)

Abstract

We investigate the One-shot Optimization strategy introduced in this form by Hamdi and Griewank for the applicability and efficiency to identify parameters in models of the earth's climate system. Parameters of a box model of the North Atlantic Thermohaline Circulation are optimized with respect to the fit of model output to data given by another model of intermediate complexity. Since the model is run into a steady state by a pseudo time-stepping, efficient techniques are necessary to avoid extensive recomputations or storing when using gradient-based local optimization algorithms. The One-shot approach simultaneously updates state, adjoint and parameter values. For the required partial derivatives, the algorithmic/automatic differentiation tool TAF was used. Numerical results are compared to results obtained by the BFGS and L-BFGS quasi-Newton method.

Keywords: Algorithmic differentiation; bounded retardation; climate model; fixed point iteration; parameter identification.

AMS Classification: 49M29; 90C30; 90C53; 92-08

1 Introduction

Parameter optimization is an important task in all kinds of climate models or models that simulate parts of the climate system, as for example ocean or atmospheric models. Still, some processes are not well-known, some are too small-scaled in time or space, and others are just beyond the scope of the model. All these processes are *parameterized*, i.e. simplified model functions (parameterizations) are used. These necessarily include lots of – most of the time – only heuristically known parameters. A main task thus is to *calibrate* the models by optimizing the parameter w.r.t. data from measurements or other (more complex) models.

Similar to many applications in engineering applications of fluid mechanics, also in geophysical flows (e.g. ocean models) an optimization is at first performed for steady states of the equations before proceeding to transient problems. This means that only the stationary solution is used in the cost or objective function to be minimized. Moreover (and this is the second point where engineering and geophysical flow problems are similar), the computation of steady states is often performed by running a transient model into the steady state. This strategy is called pseudo time-stepping, since the time variable may be regarded as a kind of iteration counter.

It is well known from optimal control of differential equations that the classical adjoint technique (that allows the representation of the gradient of the cost) leads to a huge amount of recomputations, storing or both. This problem looks even more frustrating in the pseudo-time stepping context, since here only the final, numerically converged state is important for the cost. Nevertheless a classical adjoint technique would need all intermediate iterates.

If the number of parameters to be optimized is small, a sensitivity equation approach is also reasonable. On the discrete level this is comparable to the application of the forward mode of Automatic or Algorithmic Differentiation (AD). Here, the sensitivity equation has the same temporal integration direction (namely forward) as the original pseudo time-stepping. But nevertheless it is worthwhile investigating how the two (for a non-linear model) coupled iterations for state and sensitivity are performed.

Griewank described in [1] the differences between two-phase (where the iteration for the state is run to the steady state or fixed point first, and then the sensitivity is computed) and piggy-back approaches (where both iterations are combined to one). Christianson in [2] proposed to perform the sensitivity iteration with the converged state instead of using its iterates. Giering, Kaminski and Vossbeck in [3] used the so-called *Full Jacobian approach*, where they directly used the steady state equation and differentiated it to obtain an equation for the gradient.

The approach used here is called One-shot approach, which was in this form developed by Hamdi and Griewank, and can be seen as an extension of the piggy-back strategy aiming for optimality and feasibility simultaneously with the so-called bounded retardation. That means that the number of One-shot iterations shall not too much exceed the number of fixed point iteration steps that are necessary for the computation of feasible states itself. Theoretical results were published in [4],[5], an engineering application was presented by Özkaya and Gauger in [6].

The idea of simultaneous solution of state equations and parameter correction is not new. In [7], S. Ta'asn uses a pseudo-time embedding for the state and adjoint state equations and the design equation is solved as an additional boundary condition. This still results in a differential algebraic equation which requires some strategy to solve the design equation alone.

In [8], the authors construct a system of only ODEs which is solved by a time-stepping method in the spirit of reduced SQP-methods. They develop a preconditioner working on the whole system of equations with state, costate and design equations.

In the One-shot approach used here, the idea is that for fixed parameters there is a given (not necessarily (pseudo-) time-stepping) strategy to solve the state equations. This strategy is assumed to demand no or disallow any changes. In each iteration step the update of the state is augmented by an update of the adjoint state and a kind of quasi-Newton step for the design correction with the distinctive feature that the required preconditioner controls convergence of the whole system. Here, the preconditioner is a squared matrix of only the size of the number of parameters.

Since the assumptions in the theoretical analysis of the One-shot method are very strict and the computation of the preconditioner seems at first glance laborious and expensive, the intention of this paper is to check the applicability of the One-shot strategy for real world problems and possibly propose simplifications. We compare numerical results to the gradient based BFGS and limited-memory BFGS (L-BFGS) methods. We set aside the comparison to genetic or so-called intelligent search algorithms, see e.g. [9], because the aim of the One-shot approach according to the authors of [4] and [5] is to offer an alternative to local gradient-based optimization techniques. Genetic algorithms usually demand a high number of function evaluations which we want to avoid because of the costly computation of steady states needed for the function evaluation.

In this paper, we apply the One-shot approach to a box model of the North Atlantic. This problem is different from the application in [6] in that the parameters enter in a nonlinear fashion resulting in so-called non-separable adjoints where the adjoint is no longer only the sum of a term on the state and a term on design.

The outline of this paper is the following. In section 2 we recall the One-shot optimization strategy according to [4] and [5]. We apply the One-shot method to an example in earth system modeling in section 3. There, we describe the Rahmstorf 4-box-model, the optimization problem and present numerical results. Section 4 draws conclusions.

2 One-shot Optimization Strategy

In this section, we recall the One-shot optimization strategy according to [4] and [5], its quintessence and difference to conventional optimization methods, and we derive and explain the One-shot iteration step. First of all, we describe the mathematical problem behind the parameter optimization problem.

2.1 Problem Formulation

Parameters u of a model describing physical, biological, chemical or other real life phenomena are usually determined by fitting model output $y = y(u)$ to observed data denoted by y_{data}. This data can also be taken from other, more comprehensive models.

The fitting procedure then is a mathematical optimization problem with a least-squares cost functional with some regularization term

$$J(y,u) = \frac{1}{2}\|y - y_{data}\|_2^2 + \frac{\alpha}{2}\|u - u_{guess}\|_2^2, \alpha \in \mathbb{R}_0^+$$

under the constraint that model equations, namely $c(y,u) = 0$, are fulfilled.

In climate modeling, model equations are usually partial and/or ordinary differential equations solved by an iterative process.

The problem will become more difficult with respect to uniqueness of minima and computation of derivative information, if the quantity to be fit to data g_{data} is computed from a functional $g(y,u)$ such that J then is

$$J(y,u) = \frac{1}{2}\|g(y,u) - g_{data}\|_2^2 + \frac{\alpha}{2}\|u - u_{guess}\|_2^2.$$

In the finite dimensional case or the discretized version, where $y \in Y \subset \mathbb{R}^n$, $u \in U \subset \mathbb{R}^m$ and $g : Y \times U \to \mathbb{R}^l$, the cost function is the sum of the squared differences

$$J(y,u) = \frac{1}{2}\sum_{i=1}^{l}(g_i(y,u) - g_{i,data})^2 + \frac{\alpha}{2}\sum_{i=1}^{m}(u_i - u_{i,guess})^2.$$

Here, the objective function J is $J : Y \times U \to \mathbb{R}$, $y \in Y$ is the state, $u \in U$ is the design or parameter vector to be optimized. With the help of the regularization term $\frac{\alpha}{2}\|u - u_{guess}\|_2^2$ parameters u are kept in an acceptable or presumed range around parameter values u_{guess}, where elements $u_{i,guess}$ can for example be taken as mean values of some maximum and minimum values. We assume J to be $C^{2,1}$, which means twice continuously differentiable in y and once in u. We further assume the Jacobian of c with respect to y, denoted c_y, to always be invertible, such that with the mean value theorem, there exists only one y^* with $c(y^*,u) = 0$ for a fixed u.

2.2 One-shot Iteration and its Properties

In practice, finding an analytical solution for a feasible state y^* with $c(y^*,u) = 0$ is often impossible. That is why usually an iterative method is called upon.

For the One-shot strategy, we assume that there is a given fixed point iteration, also called model spin-up , which has already been found reliable and successful in the search for the feasible state y^* for parameters u. Included step size or preconditioner strategies can be carried over and do not have influence on the One-shot iteration. Thus, there is a *given* contraction, (pseudo-) time-stepping strategy or fixed point iteration G, where y^* satisfies $y^* = G(y^*,u) = \lim_{k\to\infty} G(y_k,u)$.

The fundamental idea of the One-shot approach is to *reformulate* the condition $c(y,u) = 0$ into the *fixed point equation* $y = G(y,u)$. The iteration function $G : Y \times U \to Y$ is assumed to be $C^{2,1}$ with the contraction factor $\rho < 1$, i.e. for a suitable inner product norm $\|\cdot\|$ we have for G_y, denoting the Jacobian of G with respect to y, that

$$\|G_y(y,u)\| \leq \rho < 1, \qquad \forall y \in Y. \tag{1}$$

from which follows

$$\|G(y_1,u) - G(y_2,u)\| \leq \rho\|y_1 - y_2\|, \forall y_1, y_2 \in Y. \tag{2}$$

With the contraction property of G we can infer from the Banach fixed point theorem, for fixed u, the sequence $y_{k+1} = G(y_k, u)$ converges to a unique limit $y^* = y^*(u)$.
The assumptions on the model function c and the contraction G are very strict and rarely analytically or even numerically provable. However, we will see in our numerical example, that the One-shot strategy even converges under weaker assumptions on the contraction G. Here, in our example of the 4-box-ocean-model only the *Ciric* or *quasi-contraction* property, see [10], is fulfilled.
With the help of the fixed point reformulation, the optimization problem can be written as

$$\min_{y,u} J(y,u) \quad s.t. \quad y = G(y,u). \tag{P}$$

A conventional optimization strategy performs the following steps:

In the outer loop do in the k-th iteration step:

- *Perform a complete model spin-up (inner loop) with parameter values u_k and obtain an admissible state $y_k = y^*(u_k) = \lim_{l\to\infty} G(y_l, u_k)$.*
- *Compute the value of the cost function $J(y_k, u_k)$.*
- *Adjust model parameters obtaining u_{k+1}.*

End the outer loop when a sufficient optimality condition is satisfied.

Of course, adjusting the parameters demands further full model spin-ups and/or expensive derivative information for whose computation again full model spin-ups are necessary.
The main idea of the One-shot strategy is to adjust model parameters already during the model spin-up.
Using the method of Lagrange Multipliers, in the finite dimensional case, the associated Lagrangian to problem (P) with the Lagrange multiplier or *adjoint state* $\bar{y} \in \bar{Y}$ is

$$\begin{aligned} L(y,\bar{y},u) &= J(y,u) + \bar{y}^\top (G(y,u) - y) \\ &= N(y,\bar{y},u) - \bar{y}^\top y, \end{aligned}$$

where we introduce the shifted Lagrangian N as

$$N(y,\bar{y},u) \quad := \quad J(y,u) + \bar{y}^\top G(y,u).$$

A Karush-Kuhn-Tucker (KKT) point $(y^*, \bar{y}^*, u^*)$ fulfilling the first order necessary optimality condition must satisfy

$$\left.\begin{aligned} 0 = \tfrac{\partial L}{\partial y} &= N_y(y^*,\bar{y}^*,u^*) - \bar{y}^{*\top} \\ &= J_y(y^*,u^*) + \bar{y}^{*\top} G_y(y^*,u^*) - \bar{y}^{*\top}, \\ 0 = \tfrac{\partial L}{\partial \bar{y}} &= G(y^*,u^*) - y^*, \\ 0 = \tfrac{\partial L}{\partial u} &= N_u(y^*,\bar{y}^*,u^*) \\ &= J_u(y^*,u^*) + \bar{y}^{*\top} G_u(y^*,u^*). \end{aligned}\right\} \tag{3}$$

Motivated by this system of equations, the following coupled full step iteration, called *One-shot strategy* according to the authors of [4], [5], to reach a KKT point is derived:

Do in the k-th iteration step:

$$\left.\begin{aligned} y_{k+1} &= G(y_k,u_k), \\ \bar{y}_{k+1} &= N_y(y_k,\bar{y}_k,u_k)^\top \\ u_{k+1} &= u_k - B_k^{-1} N_u(y_k,\bar{y}_k,u_k) \end{aligned}\right\} \tag{4}$$

until there is (numerically) no change in $(y_k, \bar{y}_k, u_k)$.
Here, B_k is a design space preconditioner which must be selected to be symmetric positive definite. As mentioned above, we do not want to introduce additional preconditioners for the updates of y and

$\bar{y}$, because of the assumption that the model spin-up has already been found reliable and successful in the search for steady states.

The contractivity (2) ensures that the first equation in the coupled iteration step (4) converges ρ-linearly for fixed u. Although the second equation exhibits a certain time-lag, it converges with the same asymptotic R-factor (see [11]). As far as the convergence of the coupled iteration (4) is concerned, the goal is to find B_k that ensures that the spectral radius of the coupled iteration (4) stays below 1 and as close as possible to ρ. In subsection 2.3, we recall the formula of appropriate preconditioners B_k according to the authors of [4], [5].

Required Derivatives and Automatic Differentiation

For the One-shot update (4) and also later in the computation of the preconditioners B_k, a lot of derivative information is needed. The costs for its calculation are small compared to those of a conventional approach, because they only depend on the previous iteration step value. The storing/recomputation of intermediate partial derivatives, as for example $\frac{\partial y}{\partial u}$ for the computation of derivatives of J or N with respect to u, is not necessary which is one of the main differences and advantages compared to traditional optimization techniques.

Applying a tool for automatic/algorithmic differentiation (AD) can even more reduce costs and most importantly, AD computes *exact* derivatives without any approximation errors.

AD is a software technology to compute the derivative of a function at costs of only a small multiple of the costs for the evaluation of the function itself. With the help of source code transformation or operator overloading an AD tool provides the user with a computer programme containing the derivatives.

Those tools are for example TAF or ADiMat, which use the source code transformation approach to generate Fortran or Matlab subroutines to calculate function values and derivative information in one call, see [12] and [13], or for example ADOL-C using the operator overloading concept in C/C++ codes, see [14].

Regarding the One-shot optimization strategy, we need gradients (namely J_y, J_u) and vector-Jacobian-products which can cheaply be obtained with the reverse mode of AD. For the calculation of the preconditioner B also second derivatives and full Jacobians are needed which are calculated via the application of the reverse mode first and the forward mode afterwards. In our testings, we apply the (commercial) AD tool TAF for Fortran subroutines.

2.3 Preconditioner B and the Doubly Augmented Lagrangian

In this section, we explain the choice of the preconditioners B_k according to [4] and [5]. For the sake of simplicity, we omit the iteration index k using the notation B.

For the derivation of the preconditioner B, we introduce the doubly augmented Lagrangian L^a

$$L^a(y,\bar{y},u) \quad = \quad \frac{\alpha_L}{2}\|G(y,u)-y\|^2 + \frac{\beta_L}{2}\|N_y(y,\bar{y},u)^\top - \bar{y}\|^2 + N(y,\bar{y},u) - \bar{y}^\top y,$$

which is the Lagrangian of the original problem augmented by the errors in primal and dual feasibility. Here $\alpha_L > 0$ and $\beta_L > 0$ are weighting coefficients.

The authors of [4] prove that under certain conditions on α_L and β_L (see below), stationary points of problem (P) are also stationary points of L^a and that L^a is an exact penalty function. This leads to the idea to choose B as an approximation to the Hessian of L^a, i.e. $B \approx \nabla_{uu}L^a$.

In [4], it is proven that descent of the augmented Lagrangian is provided for any preconditioner B fulfilling

$$B \succeq B_0 := \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu}) \tag{5}$$

i.e. $B - B_0$ is positive semidefinite, and where

$$\sigma := 1 - \rho - \frac{(1+\frac{\|N_{yy}\|}{2}\beta_L)^2}{\alpha_L\beta_L(1-\rho)}. \tag{6}$$

The authors of [4] propose to choose α_L and β_L such that B_0^{-1} is as large as possible. Using (5) we get

$$\begin{aligned}\|B_0\|_2 &= \frac{1}{\sigma}\|\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu}\|_2 \\ &\leq \frac{1}{\sigma}(\alpha_L\|G_u\|_2^2 + \beta_L\|N_{yu}\|_2^2).\end{aligned}$$

Minimizing the right most formula as a function of α_L and β_L and replacing σ with (6) yields: Under the assumption that $\sqrt{\alpha_L\beta_L}(1-\rho) > 1 + \frac{\beta_L}{2}\|N_{yy}\|$ holds and $\|N_{yy}\| \neq 0$ we obtain

$$\begin{aligned}\beta_L &= \frac{3}{\sqrt{\|N_{yy}\|^2 + 3\frac{\|N_{yu}\|^2}{\|G_u\|^2}(1-\rho)^2} + \frac{\|N_{yy}\|}{2}} \quad \text{and} \\ \alpha_L &= \frac{\|N_{yu}\|^2\beta_L(1 + \frac{\|N_{yy}\|}{2}\beta_L)}{\|G_u\|^2(1 - \frac{\|N_{yy}\|}{2}\beta_L)}.\end{aligned}$$

As mentioned above, we pursue to $B \approx \nabla_{uu}L^a$. It turns out that at a stationary point of L^a, where primal and dual feasibility hold, the Hessian of L^a, namely $\nabla_{uu}L^a$, is

$$\nabla_{uu}L^a = \alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu}.$$

As L^a is an exact penalty function, we have $\nabla_{uu}L^a \succ 0$ in a neighbourhood of the constrained optimization solution. Assuming that $N_{uu} \succ 0$ implies that the preconditioner

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu}) \tag{7}$$

fulfills (5) and thus the step $\Delta u_k = -B^{-1}N_u(y_k, \bar{y}_k, u_k)$ of the coupled iteration (4) yields descent on L^a.

2.3.1 BGFS Update to avoid Computation of Full Jacobians and 2nd Order Derivatives

In the calculation of the preconditioner B full Jacobians and second derivatives are needed. On the one hand, those can also be calculated by algorithmic differentiation, but on the other hand, a possibility to avoid this is the application of a Low-Rank BFGS update to update the inverse approximation H_k of B_k. In view of the relation $B \approx \nabla_{uu}L^a$, we use the following secant equation in the update of H_k: $H_{k+1}R_k = \Delta u_k$, where

$$R_k := \nabla_u L^a(y_k, \bar{y}_k, u_k + \Delta u_k) - \nabla_u L^a(y_k, \bar{y}_k, u_k).$$

Imposing to the step multiplier η to satisfy the second Wolfe condition

$$\Delta u_k{}^\top \nabla_u L^a(y_k, \bar{y}_k, u_k + \eta\Delta u_k) \geq c_2 \Delta u_k{}^\top \nabla_u L^a(y_k, \bar{y}_k, u_k)$$

with $c_2 \in [0,1]$, implies the necessary curvature condition

$$R_k{}^\top \Delta u_k > 0. \tag{8}$$

A simpler procedure could skip the update whenever (8) does not hold by either setting H_{k+1} to identity or to the last iterate H_k. Provided (8) holds, we use

$$H_{k+1} = (I - r_k\Delta u_k R_k{}^\top)H_k(I - r_k R_k \Delta u_k{}^\top) + r_k \Delta u_k \Delta u_k{}^\top$$

with $r_k = \frac{1}{R_k{}^\top \Delta u_k}$.

The weights α_L, β_L of L^a require norms of second order derivatives. In [5], the authors propose simpler approximations according to two different approaches. In the first version then

$$\alpha_L = \frac{2\|N_{yy}\|_2}{(1-\rho)^2} \quad \text{and} \quad \beta_L = \frac{2}{\|N_{yy}\|_2},$$

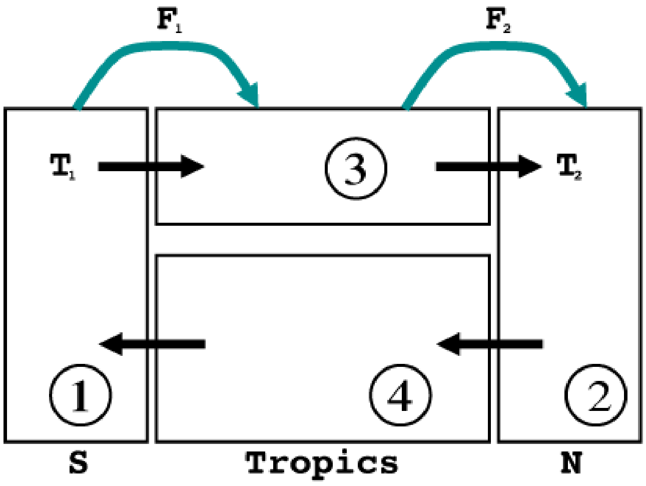

Figure 1: Rahmstorf box model, flow direction shown for $m > 0$.

in the second approach

$$\alpha_L = \frac{6\|N_{yy}\|_2}{(1-\rho)^2} \quad \text{and} \quad \beta_L = \frac{6}{\|N_{yy}\|_2}.$$

$\|N_{yy}\|_2$ can be computed via the power iteration.
For the BFGS update, the calculation of R_k requires a pure design step (step with fixed primal and dual variables y and $\bar{y}$ respectively), which might be computed at high costs. We will pay attention to this fact in our numerical example.

3 Application in Earth System Modeling

To exemplify the benefit of the One-shot optimization strategy in the case of climate research, we present the application to a 4-box-model of the Atlantic Thermohaline Circulation. The 4-box-model described in [15] simulates the flow rate of the Atlantic Ocean known as the 'conveyor belt', carrying heat northward and having a significant impact on climate in northwestern Europe. Temperatures T_i and salinity differences S_i in four different boxes $i = 1, ..., 4$, namely the southern, northern, tropical and the deep Atlantic, are the characteristics inducing the flow rate. The surface boxes exchange heat and freshwater with the overlying atmosphere, which causes a pressure-driven circulation, compare figure 1.

In [16] a smooth coupling of the two possible flow directions is proposed. The resulting time dependent ODE system reads:

$$\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m^+}{V_1}(T_4 - T_1) + \tfrac{m^-}{V_1}(T_3 - T_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m^+}{V_2}(T_3 - T_2) + \tfrac{m^-}{V_2}(T_4 - T_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m^+}{V_3}(T_1 - T_3) + \tfrac{m^-}{V_3}(T_2 - T_4)\\
\dot{T}_4 &= \tfrac{m^+}{V_4}(T_2 - T_4) + \tfrac{m^-}{V_4}(T_1 - T_4)\\
\dot{S}_1 &= \tfrac{S_0 f_1}{V_1} + \tfrac{m^+}{V_1}(S_4 - S_1) + \tfrac{m^-}{V_1}(S_3 - S_1)\\
\dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m^+}{V_2}(S_3 - S_2) + \tfrac{m^-}{V_2}(S_4 - S_2)\\
\dot{S}_3 &= \tfrac{S_0(f_2 - f_1)}{V_3} + \tfrac{m^+}{V_3}(S_1 - S_3) + \tfrac{m^-}{V_3}(S_2 - S_4)\\
\dot{S}_4 &= \tfrac{m^+}{V_4}(S_2 - S_4) + \tfrac{m^-}{V_4}(S_1 - S_4)
\end{aligned}$$

where for some positive a, $m^+ = \frac{m}{1-e^{-am}}$ almost coincides with the meridional volume transport or overturning

$$m = k(\beta_m(S_2 - S_1) - \alpha_m(T_2 - T_1))$$

for positive m and is almost zero for negative m. The term $m^- = \frac{-m}{1-e^{am}}$ becomes almost zero for positive m and $-m$ for negative m. That means the summands including m^+ and m^- are activated or deactivated depending on the flow direction. The deviation from the physically correct model becomes smaller the larger a is. Several model parameters are involved, the most important being the freshwater flux f_1 containing atmospheric water vapor transport and wind-driven oceanic transport; they are used to simulate global warming in the model and are chosen in the interval $[-0.2, 0.15]$. T_i^*, $i = 1, 2, 3$ are so-called restoring temperatures, which can be seen as counterparts of the three surface temperatures. Further model parameters are physical, relaxation and coupling constants among which there are well-known fixed parameters and those which are tunable parameters. See [15] for an explanation of the occurring constants, fixed parameters and tunable parameters.

3.1 The Optimization Problem

As mentioned in the introduction, in climate modeling an optimization is at first performed for steady states, which means in this example for temperatures and salinities which do not change in time anymore. Given fresh water fluxes $(f_{1,i})_{i=1}^l$, corresponding to different warming scenarios, the aim is to fit the overturning values $m_i = m(y(f_{1,i}), u)$ computed from stationary temperatures and salinities $(T_1, T_2, S_1, S_2)_i$ obtained by the model spin-up for $f_{1,i}$ to data $m_{d,i}$ from a more complex model *Climber2*, see [17]. $u = (T_1^*, T_2^*, T_3^*, \Gamma, k, a)$ are the control parameters. Here, Γ is a thermal coupling constant in the computation of the thermal relaxation constants λ_i, $i = 1, 2, 3$. All other parameters occur in the model description of the previous subsection. Using notations from section 2, the state is $y = (y_i)_{i=1}^l$ with $y_i = y(f_{1,i}) = (T_1, T_2, T_3, T_4, S_1, S_2, S_3, S_4)_i$.

If $F(y, u)$ denotes the right-hand side of the ODE system of the model, we get

$$\begin{aligned} \min_{y,u} J(y,u) \quad &:= \quad \frac{1}{2}\|m(y(f_1),u) - m_d\|_2^2 + \frac{\alpha}{2}\|u - u_{guess}\|_2^2, \\ \text{s.t.} \quad 0 \quad &= \quad F(y(f_{1,i}),u), \quad i = 1, ..., l. \end{aligned}$$

The regularization term incorporates a prior guess u_{guess} for the parameters. The larger α the more the parameters u are kept close to u_{guess}.

The difficulty here is that $m : \mathbb{R}^{8l} \times \mathbb{R}^6 \to \mathbb{R}^l$ is not injective. There are several combinations of steady/feasible T_1, T_2, S_1, S_2 and the parameter $u(5) = k$ to compute the same overturning m. The smaller α the more likely the different optimization strategies find completely different optimal parameters with almost the same function values $J(y^*, u^*)$.

In [15] the authors apply the Explicit Euler time stepping with a fixed step size of one year, i.e. $\Delta t = 1$, to run the model into a steady state. Otherwise, known model constants scaled on a time span of one year must be adjusted. Thus G defined in section 2 here represents one full Euler step $y_{k+1} = G(y_k, u) = y_k + F(y_k, u)$ operating on all freshwater fluxes $f_{1,i}$ together, i.e. for fixed u we have $G(\cdot, u) : \mathbb{R}^{8l} \to \mathbb{R}^{8l}$.

In this example, contractivity of G is not given in general, i.e. ρ in (1) exceeds 1 for several steps. However, in average it is less than 1. Here, for the explicit Euler sequence $y_{k+1} = G(y_k) = y_k + F(y_k, u)$, the quasi-contraction property [10]

$$\|y_{k+1} - y_k\| \leq q \max\{\|y_k - y_{k-1}\|, \|y_{k+1} - y_{k-1}\|\}$$

for $0 \leq q < 1$ holds. In our testings, G converges for fixed u but different starting values y_0 to the same stationary y^*.

3.2 Numerical Results and Discussion

In our numerical testing, we compare the two versions of the One-shot method, with full computation of the preconditioner B on the one hand and BFGS update of B on the other hand, to a traditional BFGS-quasi-Newton optimization approach. Furthermore, we compare results to values obtained by the Limited-memory BFGS (L-BFGS) algorithm implemented by Zhu, Byrd, Nocedal and Morales, see [18], version 3.0 from 2011, without and finally with box constraints on the control parameters (L-BFGS-B) because we find that computed optimal parameter values of the BFGS and L-BFGS method are far away from actual real world values. In the three different BFGS approaches, for each parameter value u_k during the optimization process the box model has to be run into a steady state. In our example, that takes between 4,000 and 15,000 Euler steps. Compared to more complex climate models, here the Euler time step evaluation is not expensive. However, during the optimization process a large number of Explicit Euler time steps will accumulate and for derivative calculation a huge amount of recomputations, storing or both is necessary. That becomes obvious in the calculation of derivatives using automatic differentiation. Whereas for the BFGS method in the reverse mode it is necessary to store all Euler steps until a steady state is reached, in the One-shot method the required derivatives depend on the current values only, i.e. on only one Euler step.

In our implementation we replaced $u_i = T_i^*$, $i = 1, ..., 3$, with $\tilde{u}_i = W_i$ such that $u_i = \tilde{u}_i + u_{fix}$ where $(u_{fix})_1^3 = (6.64, 2.68, 11.69)$, which are optimal values calculated in [15]. W_i, $i = 1, ..., 3$, can be interpreted as warming trends. We chose $u_{guess} = (0., 0., 0., 23., 25., 500)$ as starting parameters. Since only quasi-contraction is given, we expect the contraction factor ρ to exceed 1 for several iteration steps possibly resulting in arithmetic exceptions. Therefore, we fix ρ close to 1, namely $\rho = 0.9$.

For better initialization especially of the adjoint, we propose an update of only the state and adjoint state for the first 500 iteration steps.

The One-shot-BFGS strategy demands a linesearch procedure, otherwise the method fails. Here, we applied a simple strategy constantly halfing the steplength until there is a reduction in the costfunction with the resulting step.

We perform our numerical testings on a SUN-W-Ultra-SPARC-IIIi CPU 1.3GHz machine.

3.2.1 Influence of Rare Update of Weighting Coefficients of the Preconditioners B_k on the Optimization

In the first version, we calculate preconditioners B_k defined in (7) in every iteration including all first and second order derivatives. Also the weighting coefficients α_L, β_L and σ are adjusted. We find, that the weights do not change significantly from iteration to iteration. As one can see in Table 1, an update performed only after several time-steps does not significantly influence the optimization but the computational time needed. Therefore, we prefer the version with a calculation of α_L, β_L and σ every 1,000 iterations.

Table 1: Effect on the optimization of rare update of the weights α_L, β_L and σ for $\alpha = 0.1$. We compare the values of the cost functional, the weighted data fit, the number of iterations and the computational time in minutes.

update of weights of B	$J(y^*, u^*)$	data fit	# iterations	comp.time
every 10,000 iterations	14.544	0.399	1,185,500	5.085
every 1,000 iterations	14.544	0.399	1,182,053	5.067
every iteration	14.544	0.399	1,181,701	10.045

3.2.2 Effect of the Weighting Factor α on the Numerical Results

In the following, our attention is drawn on the effect of the weighting factor α in front of the penalty term $\|u - u_{guess}\|$. For the last parameter a we chose the additional factor 0.01, because a is of higher dimension than the other parameters and can vary more. Here in the example of the 4-box-model, without any regularization, i.e. $\alpha = 0$, the One-shot method and the L-BFGS method without constraints do not converge or fail. The BFGS method and the L-BFGS with box constraints terminate with parameter values u^* where $\|J_u(y(u^*), u^*)\|$ still is very large, but the algorithms cannot find descent directions.

We recall from section 3.1 that the considered optimization problem has several local minima which might be of the same quality regarding the data fit, even though the obtained model parameters are completely different. The larger the weighting factor α the less the obtained parameters vary.

In our testings with $\alpha > 0$, we compare the optimal value of the cost function, the data fit weighted to the number of observations, the number of iteration steps, the number of needed Euler steps, and the computational time in minutes. Furthermore, we take a look at the quality of optimality, which means for the Oneshot strategies the norm of $L_{(y,\bar{y},u)}(y^*, \bar{y}^*, u^*)$ and for the BFGS methods the norm of $J_u(y(u^*), u^*)$. The numerical results are collected in tables 2 and 3 and illustrated in figures 2 and 3.

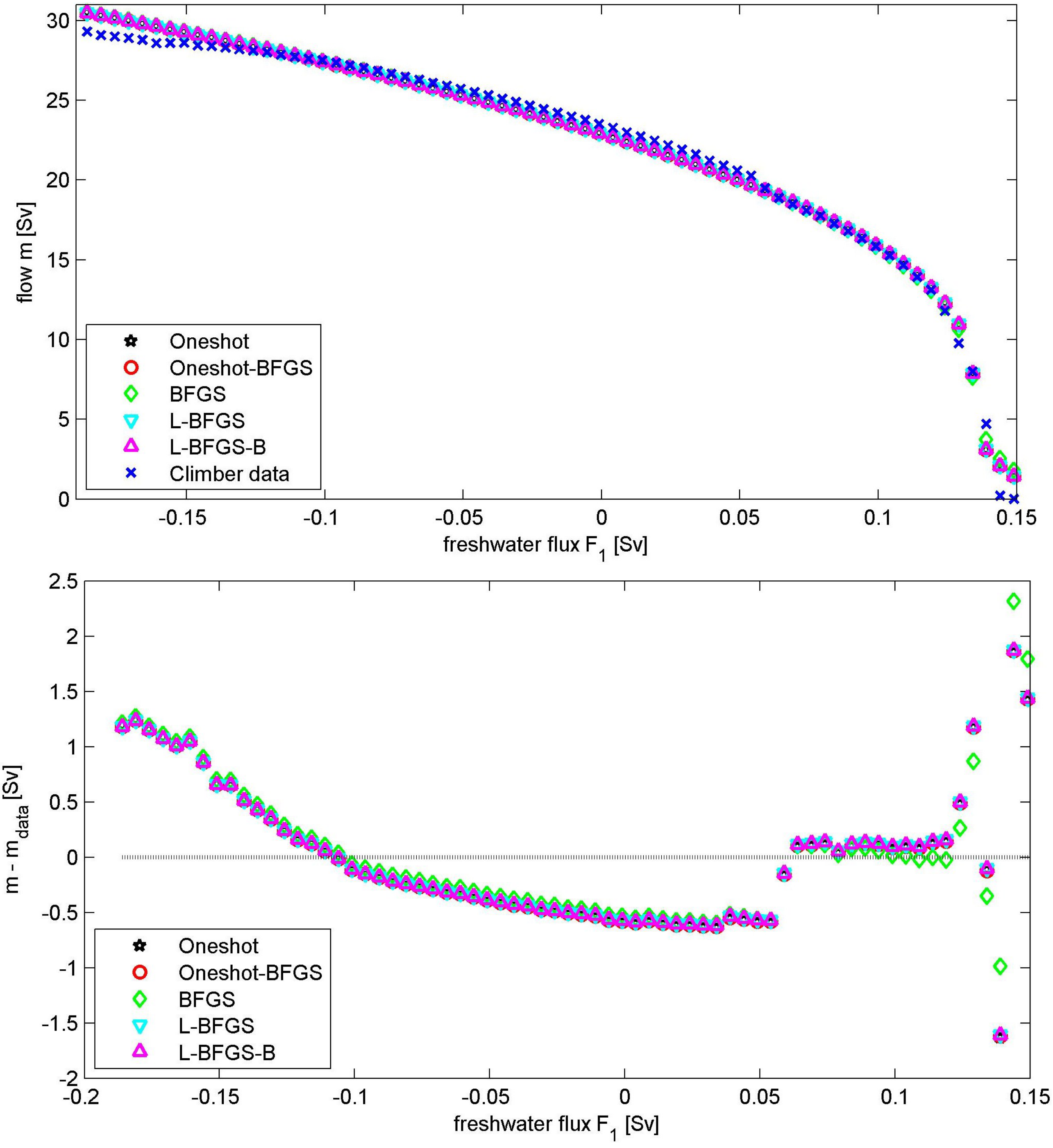

Figure 2: Results of the optimization comparing the One-shot strategies with the BFGS-quasi-Newton methods for $\alpha = 0.1$ (left) and the differences to the Climber data (right).

Not surprisingly, one generally detects that the smaller α the better the fit of data becomes.
We observe that for different α the qualities of the methods vary. Especially for large fresh water fluxes $f_{1,i}$ the outputs of the different optimization strategies strongly differ. These are $f_{1,i}$ for which the model switches the flow direction of m during the model spin-up.
Comparing the original One-shot and the One-shot-BFGS methods, the presumption that the One-

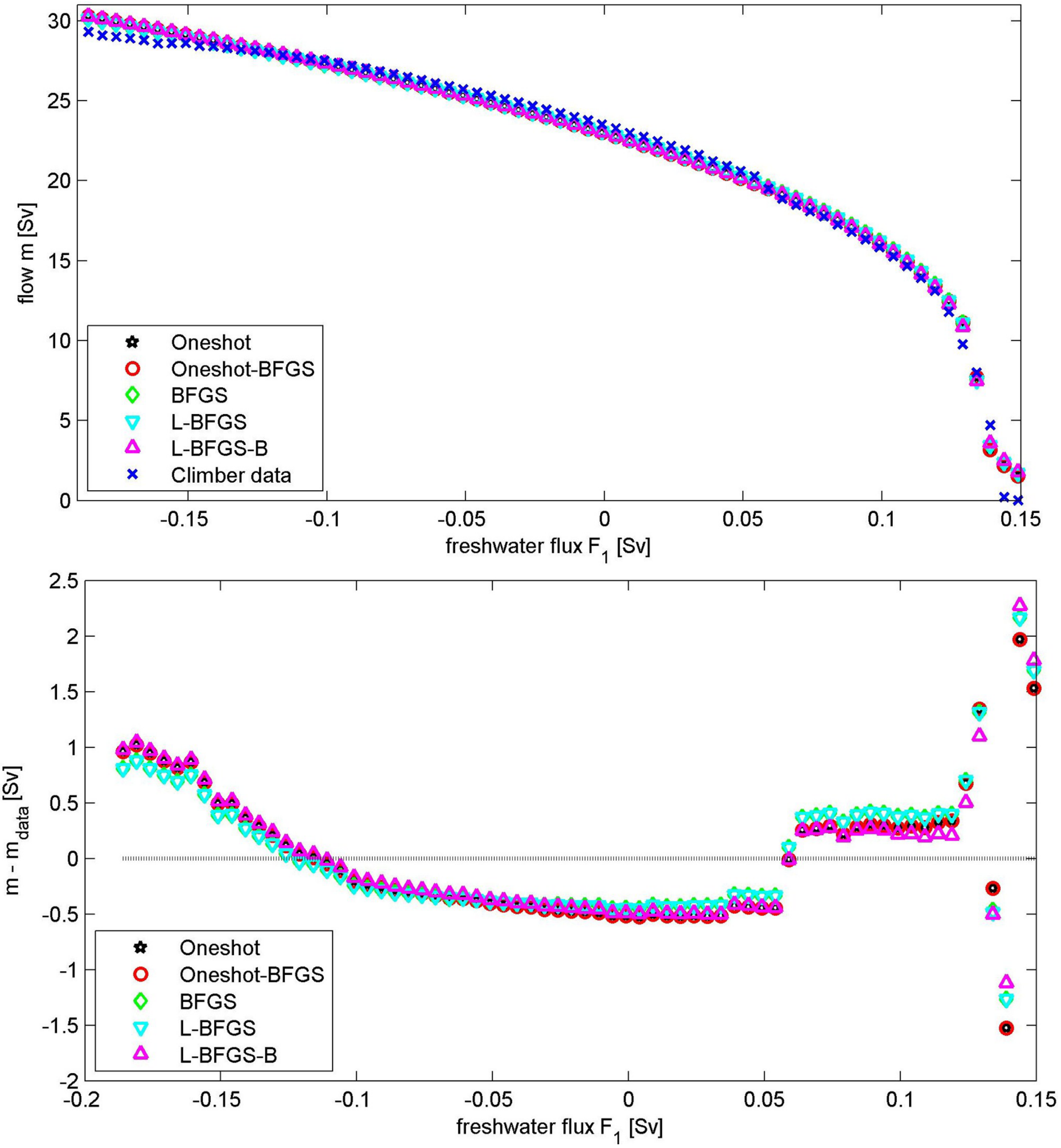

Figure 3: Results of the optimization comparing the One-shot strategies with the BFGS-quasi-Newton method for $\alpha = 0.001$ (left) and the difference to the Climber data (right).

shot-BFGS strategy might be rather time consuming due to the additional pure design steps is confirmed. Here, in an example with a very small number of parameters to be optimized, the One-shot-BFGS approach is not recommended. However, in problems with a large number of design variables, the One-shot-BFGS approach might be an alternative. The computed data fit can be regarded as equally good in this example.

For $\alpha = 10$ the strategies show almost no difference in their results, neither in the fit of the data nor in the computed optimal parameters. Concerning computational time and the number of Euler steps, the original Oneshot strategy perform best.

For $\alpha = 1$ and $\alpha = 0.1$ the One-shot strategy shows difficulties in performance. We suspect that here the balance between keeping parameters close to u_{guess} and reducing the misfit has a disadvantageous influence on the One-shot method. However, also the BFGS method does not perform well for $\alpha = 0.1$.

Table 2: Results of the optimization comparing values of the cost functional, the weighted data fit, the number of iterations of the optimization procedure, the number of needed Euler steps, the quality of optimality ($\|L_{(y,\bar{y},u)}(y^*,\bar{y}^*,u^*)\|$ for methods 1-2 and $\|J_u(y(u^*),u^*)\|$ for methods 3-5 respectively) and the computational time in minutes

	Method	$J(y^*,u^*)$	data fit	#iterations	#Euler steps	opt.cond.	comp.time
$\alpha = 10$	1	25.810	0.539	254,859	254,859	2.0E-1	1,086
	2	25.810	0.539	220,687	220,687	1.5E-1	1.864
	3	25.809	0.539	41	761,661	2.5E-5	1.715
	4	25.809	0.539	31	411,732	5.5E-4	1.351
	5	25.809	0.539	39	540,946	2.2E-4	1.785
$\alpha = 1$	1	17.438	0.462	1,212,057	1,212,057	1.8E-1	5.155
	2	17.434	0.462	1,101,992	1,101,992	1.3E-1	9.315
	3	17.426	0.462	48	926,049	1.3E-4	2.084
	4	17.426	0.462	46	653250	3.3E-4	2.142
	5	17.426	0.462	53	1,194,483	3.2E00	3.938
$\alpha = 0.1$	1	14.544	0.399	1,182,053	1,182,053	2.3E-1	5.067
	2	14.469	0.398	3,122,016	3,122,016	1.3E-1	26.366
	3	15.571	0.401	54	1,403,864	3.1E-1	2.878
	4	14.417	0.393	71	1,250,796	2.6E-2	4.100
	5	14.417	0.393	76	1,412,728	6.0E-4	4.657
$\alpha = 0.01$	1	13.747	0.396	437,543	437,543	2.2E-1	1.917
	2	13.786	0.396	344,463	344,463	7.0E-1	2.906
	3	12.514	0.338	52	1,455,877	2.3E-1	3.137
	4	fails					
	5	13.747	0.397	29	672,388	4.1E00	2.217
$\alpha = 0.001$	1	12.232	0.352	150,410	150,410	1.4E-1	0.649
	2	12.364	0.353	170,597	170,597	6.6E-1	1.442
	3	11.411	0.331	64	2,134,827	5.9E-1	4.457
	4	11.412	0.331	63	2,018,101	3.6E-1	6.233
	5	12.257	0.352	77	2,928,986	4.2E00	9.078

Legend of Methods: 1 One-shot, 2 One-shot-BFGS, 3 BFGS, 4 L-BFGS, 5 L-BFGS-B

For smaller α, we observe significant differences. The unconstrained BFGS strategies find the best fit, but parameter values (u_1^*, u_2^*, u_3^*) which are not acceptable in this real world problem. L-BFGS-B computes similar results as the One-shot method, but needs far more Euler steps and therewith a much longer computational time.

We detect that the parameters computed by the One-shot method stay in acceptable ranges without any box constraints. Computed parameters are to some extend similar to those of the L-BFGS-B method.

One main goal of the One-shot strategy was to achieve so-called bounded retardation for the speed of convergence compared to the number of time-steps needed to run the model into a steady state. Since the Explicit Euler time-stepping does not show quick converge and ratios $\theta_k = \frac{\|y_{k+1}-y_k\|}{\|y_k-y_{k-1}\|}$ even exceed 1 for several steps k, one cannot expect the One-shot method to converge very fast in this special example. The average value for θ in a pure model spin-up with parameters taken from [15] is 0.992 and for the One-shot strategy ($\alpha = 0.1$) $\theta = 0.9999884$.

Furthermore, the number of One-shot iteration steps was intended to exceed the number of Euler steps of a single model spin-up not too much. Especially for parameter sets near the computed solution, a model spin-up with fixed parameters needs 12,000 to 15,000 Euler steps. For $\alpha \in \{10, 0.01, 0.001\}$ where the One-shot strategy shows good performance, the observed number of iterations is about 10 to 40 times larger than the number of Euler steps for one single spin-up. Considering that the BFGS strategies need at least about 30 optimization steps requiring further function evaluations and model spin-ups the factor is not very large. Even in those cases, where the One-shot strategies does not show

Table 3: Optimal parameters computed by the different optimization strategies.

	Method	u_1	u_2	u_3	u_4	u_5	u_6
$\alpha = 10$	One-shot	0.482	0.457	-0.916	23.211	25.283	502.83
	One-shot-BFGS	0.480	0.457	-0.918	23.211	25.283	502.83
	BFGS	0.475	0.451	-0.927	23.209	25.280	502.83
	L-BFGS	0.475	0.451	-0.927	23.209	25.280	502.83
	L-BFGS-B	0.475	0.451	-0.927	23.209	25.280	502.83
$\alpha = 1$	One-shot	0.585	0.583	-0.957	23.492	24.902	512.44
	One-shot-BFGS	0.569	0.568	-0.977	23.488	24.893	512.42
	BFGS	0.521	0.522	-1.042	23.474	24.865	512.34
	L-BFGS	0.521	0.521	-1.042	23.474	24.865	512.34
	L-BFGS-B	0.520	0.520	-1.042	23.474	24.866	512.34
$\alpha = 0.1$	One-shot	1.342	1.376	-0.519	24.850	23.063	528.70
	One-shot-BFGS	1.189	1.238	-0.915	24.601	22.638	527.81
	BFGS	1.244	1.497	-2.741	23.524	19.831	503.50
	L-BFGS	0.752	0.816	-1.570	24.410	22.262	527.03
	L-BFGS-B	0.752	0.816	-1.570	24.410	22.262	527.03
$\alpha = 0.01$	One-shot	0.237	0.251	0.030	28.961	24.997	534.91
	One-shot-BFGS	0.184	0.286	0.079	32.577	23.732	531.10
	BFGS	2.277	2.899	-5.176	30.650	14.803	503.26
	L-BFGS	fails					
	L-BFGS-B	1.130	1.4351	-2.490	26.983	19.662	502.024
$\alpha = 0.001$	One-shot	1.280	1.652	-0.606	43.710	18.728	520.64
	One-shot-BFGS	0.423	0.764	-0.041	46.850	19.847	521.19
	BFGS	3.027	3.805	-6.832	32.289	12.999	503.93
	L-BFGS	3.023	3.800	-6.825	32.349	13.007	503.94
	L-BFGS-B	1.042	1.559	-3.000	45.568	16.741	507.95

quick convergence, the number of iterations still is not too far away from the number of Euler steps required by the BFGS strategies. In applications, where the fixed point iteration G is more expensive than the Euler time-stepping applied in this example, the One-shot strategy then might catch up with the needed computational time.

4 Conclusions

We have successfully applied the One-shot method according to Hamdi and Griewank, [4], [5], to a parameter optimization problem in ocean modeling. We have analyzed its applicability and find that the One-shot strategy presents a promising approach to optimize models consuming much time and calculational costs for their spin-ups using (pseudo-)time stepping or a fixed point iteration. Our numerical example was the parameter optimization of the Rahmstorf 4-box-model of the North Atlantic with steady states achieved via an Explicit Euler spin-up. Optimization results of the original One-shot strategy and the One-shot-BFGS method with an BFGS update of the preconditioner of the parameter correction step are compared to a classical BFGS-quasi-Newton method and the L-BFGS-method with and without box constraints on the parameters.

We observed that the One-shot-BFGS strategy does not show good performance in this example with only 6 parameters. The original version with full computation of the preconditioner performs well for large and very small weighting factors α in front of the penalty term. Further analysis on why Oneshot has difficulties in finding optimal values for weights $\alpha \in \{1, 0.1\}$ can be valuable.

We have found out that the One-shot method can be applied even though contractivity is not given in general and that fixing the contraction factor ρ to a number close to 1 is adequate. Furthermore, computation of the weights of B is not mandatory in each iteration step.

Considering examples with more expensive model spin-ups, the One-shot method might on the one hand even gain (or at least catch up in those examples with slow convergence) concerning computational time and on the other hand be the only applicable alternative for derivative based optimization methods, because derivatives depend on one spin-up step only instead of the whole spin-up, which is the main difference and advantage compared to standard methods. The application to earth system models involving nonlinear PDEs and/or a higher spatial resolution with computationally more expensive model solvers and periodic solutions will be of great interest for future investigations to demonstrate the efficiency of the One-shot approach.

References

[1] Griewank, A., *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM, Philadelphia, PA (2000).

[2] Christianson, B., Reverse accumulation and implicit functions. *Optimization Methods and Software*, 9(4), 307–322 (1998).

[3] Kaminski, T., Giering, R., and Voßbeck, M., Efficient sensitivities for the spin-up phase. Automatic Differentiation: Applications, Theory, and Implementations, Lecture Notes in Computational Science and Engineering, Springer, New York, 50, 283–291 (2005).

[4] Hamdi, A. and Griewank, A., Reduced Quasi-Newton Method for Simultaneous Design and Optimization. Comput. Optim. Appl. online, Available at `www.springerlink.com` (2009).

[5] Hamdi, A. and Griewank, A., Properties of an Augmented Lagrangian for Design Optimization. *Optimization Methods and Software*, 25(4), 645–664 (2010).

[6] Özkaya, E. and Gauger, N., Single-Step One-Shot Aerodynamic Shape Optimization. *International Series of Numerical Mathematics*, 158, 191–204 (2009).

[7] Ta’asn, S., Pseudo-Time Methods for Constrained Optimization Problems Governed by PDE. ICASE Report No. 95-32 (1995).

[8] Hazra, S. B. and Schulz, V., Simultaneous Pseudo-Timestepping for PDE-Model Based Optimization Problems. *BIT Numerical Mathematics*, 44, 457–472 (2004).

[9] Pham, D. and Karaboga, D., Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. Springer London, Limited (2012).

[10] Ciric, L. B., A Generalization of Banach’s Contraction Principle. *Proceedings of the American Mathematical Society*, 45(2), 267–273 (1974).

[11] Griewank, A. and Kressner, D., ”Time-lag in Derivative Convergence for Fixed Point Iterations. ARIMA Numéro spécial CARI’04, 87–102 (2005).

[12] Giering, R., Kaminski, T., and Slawig, T., Generating Efficient Derivative Code with TAF: Adjoint and Tangent Linear Euler Flow Around an Airfoil. *Future Generation Computer Systems*, 21(8), 1345–1355 (2005).

[13] Bischof, C. H., Lang, B., and Vehreschild, A., Automatic Differentiation for MATLAB Programs. *Proceedings in Applied Mathematics and Mechanics*, 2(1), 50–53 (2003).

[14] Griewank, A., Juedes, D., and Utke, J., Algorithm 755: ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++. *ACM Transactions on Mathematical Software*, 22(2), 131–167 (1996).

[15] Zickfeld, K., Slawig, T., and Rahmstorf, S., A low-order model for the response of the Atlantic thermohaline circulation to climate change. *Ocean Dynamics*, 54, 8–26 (2004).

[16] Titz, S., Kuhlbrodt, T., Rahmstorf, S., and Feudel, U., On freshwater-dependent bifurcations in box models of the interhemispheric thermohaline circulation. *Tellus A*, 54, 89 – 98 (2002).

[17] Rahmstorf, S., Brovkin, V., Claussen, M., and Kubatzki, C., CLIMBER-2: A climate system model of intermediate complexity. Part II: Model sensitivity. *Clim. Dyn.*, 17, 735–751 (2001).

[18] Zhu, C., Byrd, R. H., and Nocedal, J., L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4), 550–560 (1997).

Acknowledgments

The authors are grateful to the DFG for financial support within the Priority Program SPP1253 *Optimization with partial differential equations.* They would like to thank the reviewers for their valuable comments that helped to improve the manuscript.

***Claudia Kratzenstein** is a PhD student at Christian-Albrechts-University Kiel, Germany. She received her degree (Diplom) in mathematics at Humboldt University Berlin, Germany, in the field of non-linear optimisation. Among her research interests are non-linear optimisation methods, especially quasi-Newton-strategies, automatic differentiation, and their application in climate modeling.*

***Thomas Slawig** is a Professor for Algorithmic Optimal Control – Oceanic CO_2 Uptake at Christian-Albrechts-University Kiel, Germany. He received his PhD in Applied Mathematics in 1998 from TU Berlin with a thesis on Shape Optimization for Navier-Stokes Equations. Since 1999 he is working in the field of parameter identification, sensitivity analysis and numerics for climate models. His research interests are numerical mathematics and optimisation, automatic differentiation, and optimal control.*

One-shot Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation

Claudia Kratzenstein[*], Thomas Slawig[†]

(Published on the Preprint server of the DFG-SPP 1253 in September 2008; available at: http://www.am.uni-erlangen.de/home/spp1253/wiki/images/b/ba/Preprint-spp1253-11-03.pdf)

Abstract

Parameters of a box model of the north Atlantic thermohaline circulation are optimized to fit the model results, i.e. the overturning, to data given by a more detailed climate model of intermediate complexity. Since the model is run into a steady state by a pseudo time-stepping, efficient techniques are necessary to avoid extensive recomputations and/or storing when using adjoint-based gradient representations for local optimization algorithms. The preconditioned one shot approach studied by Hamdi and Griewank that simultaneously updates state, adjoint and parameter values, is applied to this nonlinear climate model. For the required partial derivatives, a software tool of Algorithmic/Automatic Differentiation was used. Numerical results are compared to results obtained by a Quasi-Newton and a gradient method.

Keywords: Parameter optimization, pseudo time-stepping, fix point solver, climate model, automatic differentiation

1 Introduction

Parameter optimization is an important task in all kind of climate models or models that simulate parts of the climate systems, as for example ocean or atmospheric models. Still, some processes are not well-known, some are too small-scaled in time or space, and others are just beyond the scope of the model. All these processes are *parameterized*, i.e. simplified model functions (parameterizations) are used. These necessarily include lots of – most of the time – only heuristically known parameters. A main task thus is to *calibrate* the models by optimizing the parameter w.r.t. to data from measurements or other (more complex) models.

Similar to many applications in engineering applications of fluid mechanics, also in geophysical flows (e.g. ocean models) an optimization is at first performed for steady states of the equations before proceeding to transient problems. This means that only the stationary solution is used in the cost or objective function to be minimized. Moreover (and this the second point where engineering and geophysical flow problems are similar), the computation of steady states is often performed by running a transient model into the steady state. This strategy is called pseudo time-stepping, since the time variable may be regarded as a kind of iteration counter.

It is well known from optimal control of differential equations that the classical adjoint technique (that allows the representation of the gradient of the cost) leads to a huge amount of recomputations, storing or both. This problem looks even more annoying in the pseudo-time stepping context, since here only the final, numerically converged state is important for the cost. Nevertheless a classical adjoint technique would need all intermediate iterates.

If the number of parameters to be optimized is small, a sensitivity equation approach is also reasonable. On the discrete level this is comparable to the application of the forward mode of Automatic or Algorithmic Differentiation (AD). Here, the sensitivity equation has the same temporal integration

[*] `ctu@informatik.uni-kiel.de`, Institut f. Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany. Research supported by DFG SPP 1253 and DFG Cluster of Excellence *The Future Ocean*.

[†] `ts@informatik.uni-kiel.de`, Institut f. Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany. Research supported by DFG SPP 1253 and DFG Cluster of Excellence *The Future Ocean*.

direction (namely forward) as the original pseudo time-stepping. But nevertheless it is worth while investigating how the two (for a non-linear model) coupled iterations for state and sensitivity are performed.

Griewank described in [3] the differences between two-phase (where the iteration for the state is run to the steady state or fixpoint first, and then the sensitivity is computed) and piggy-back approaches (where both iterations are combined to one). Christianson in [2] proposed to perform the sensitivity iteration with the converged state instead of using its iterates. Giering, Kaminski and Vossbeck in [6] used the so-called *Full Jacobian approach*, where they directly used the steady state equation and differentiated it to obtain an equation for the gradient.

The approach used here is called One-shot approach, was developed by Griewank and Hamdi, and can be seen as an extension of the piggy-back strategy. The theoretical results were published in [5],[4], an engineering application was presented by Özkayan and Gauger in [7].

In this paper we apply the One-Shot approach to an box model of the North Atlantic. We want to minimize the differences between the output of the steady state of the model compared to data given by a more complex coupled climate model. The box model is a non-linear ODE system with eight equations for temperature and salinity in four ocean compartments. As parameters to be optimized we choose here the so-called restoring temperatures, which can be seen as counterparts of surface temperatures.

The outline of this paper is the following. In the next section we describe the box model in its original version and in a modified, differentiable version that we used here. In Section 3 we present the formulation of the optimization problem and results obtained by standard optimization methods taking from MATLAB®'s optimization toolbox, namely a gradient and a Quasi-Newton method. In the fourth section we briefly review the One-shot method, which is then applied in the next section to the particular case of the box model. In Section 6 we present numerical results of the One-shot method, which are then compared to the results of the two standard optimization techniques in the last section. Here we also give some concluding remarks and an outlook to future work. In the Appendix, we sketch the proof of contractivity of the used time stepping scheme.

2 The box model of the thermohaline circulation

In this section we present the formulation of the used climate model, a so-called box model. We start with the original version by Rahmstorf, and proceed with an advanced version which provides a differentiable right-hand side of the ODE system. Moreover we describe the used time-stepping scheme.

2.1 Introduction to the 4-box model

The basic 4-box model of the Atlantic Thermohaline Circulation described by S. Rahmstorf, K. Zickfeld and T. Slawig [11] simulates the flow rate of the Atlantic Ocean known as the 'conveyor belt', carrying heat northward and having a significant impact on climate in northwestern Europe. Temperatures T_i and salinity differences S_i in four different boxes $i = 1, ..., 4$, the southern, northern, tropical and the deep Atlantic, are the characteristics inducing the flow rate. The surface boxes exchange heat and freshwater with the overlying atmosphere, which causes a pressure-driven circulation. The following nonlinear system of ordinary differential equation describes the flow for the present circulation direction.

$$\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m}{V_1}(T_4 - T_1) & \dot{S}_1 &= \tfrac{S_0 f_1}{V_1} + \tfrac{m}{V_1}(S_4 - S_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m}{V_2}(T_3 - T_2) & \dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m}{V_2}(S_3 - S_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m}{V_3}(T_1 - T_3) & \dot{S}_3 &= \tfrac{S_0(f_2 - f_1)}{V_3} + \tfrac{m}{V_3}(S_1 - S_3)\\
\dot{T}_4 &= \tfrac{m}{V_4}(T_2 - T_4) & \dot{S}_4 &= \tfrac{m}{V_4}(S_2 - S_4)\\
m &= k(\beta(S_2 - S_1) - \alpha(T_2 - T_1));
\end{aligned}$$

where m is the meridional volume transport or overturning. The constants f_1, f_2 are freshwater fluxes containing atmospheric water vapor transport and wind-driven oceanic transport. They are multiplied by a reference salinity S_0 for conversion to a salt flux. Moreover, k is a coupling constant of flow, α, β are expansion coefficients and T_i^* temperatures towards which the surface boxes $i = 1, ..., 3$ are relaxed. The λ_i are thermal coupling constants and V_i volumes of boxes $i = 1, ..., 4$. Figure 1 illustrates the flow. In [11] the authors apply an explicit Euler time stepping to run the model into a steady state. m is set to zero if a negative flow rate is calculated. That means this model simulates only the flow with an upwelling in the south and downwelling in the north, which is the present circulation direction. For the inverse direction the advective terms of the model must be reformulated.

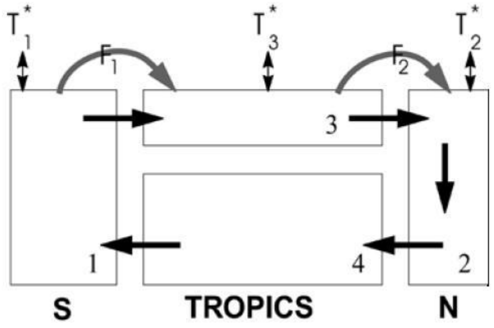

Figure 1: The Rahmstorf 4-box model

The disadvantage of the model is that it is not differentiable for $m = 0$ (or $T_1 = T_2$ and $S_1 = S_2$), but the theory of the One-shot approach used here requires a differentiable model.

2.2 The differentiable version of the box model

Titz, Kuhlbrodt, Rahmstorf and Feudel [10] propose a coupling of the two flow directions. The ODEs of the resulting advanced box model look as follows:

$$\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m^+}{V_1}(T_4 - T_1) + \tfrac{m^-}{V_1}(T_3 - T_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m^+}{V_2}(T_3 - T_2) + \tfrac{m^-}{V_2}(T_4 - T_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m^+}{V_3}(T_1 - T_3) + \tfrac{m^-}{V_3}(T_2 - T_4)\\
\dot{T}_4 &= \tfrac{m^+}{V_4}(T_2 - T_4) + \tfrac{m^-}{V_4}(T_1 - T_4)\\
\dot{S}_1 &= \tfrac{S_0 f_1}{V1} + \tfrac{m^+}{V_1}(S_4 - S_1) + \tfrac{m^-}{V_1}(S_3 - S_1)\\
\dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m^+}{V_2}(S_3 - S_2) + \tfrac{m^-}{V_2}(S_4 - S_2)\\
\dot{S}_3 &= \tfrac{S_0(f_2 - F_1)}{V_3} + \tfrac{m^+}{V_3}(S_1 - S_3) + \tfrac{m^-}{V_3}(S_2 - S_4)\\
\dot{S}_4 &= \tfrac{m^+}{V_4}(S_2 - S_4) + \tfrac{m^-}{V_4}(S_1 - S_4)
\end{aligned} \tag*{(Box)}$$

where for some positive a,

$$m^+ = \frac{m}{1 - e^{-am}}, \qquad m^- = \frac{-m}{1 - e^{am}}, \qquad m = k(\beta(S_2 - S_1) - \alpha(T_2 - T_1)).$$

The deviation from the physically correct model becomes smaller the larger a is. Using the factors m^+ and m^- the last summand nearly vanishes for a clockwise flow, i.e. $m > 0$, and the middle term nearly vanishes for the opposite direction, i.e. $m < 0$.

In the following, we denote $y = (T_1, T_2, T_3, T_4, S_1, S_2, S_3, S_4)$, $u = (T_1^*, T_2^*, T_3^*)$ and $F(y, u)$ as the right hand side of the system of ODEs of the advanced box model *(Box)*.

We used the explicit Euler time-stepping to run the model into a steady state, that means for iteration k the next state is $y_{k+1} = y_k + \Delta t F(y_k, u)$ for $\Delta t > 0$.

3 The optimization problem

In this section we formulate the optimization problem. Moreover, we give an optimization result using two black-box algorithms, namely a Quasi-Newton method with BFGS update and a gradient method, from MATLAB's optimization toolbox (routine `fminunc`).
Our goal was to optimize the advanced model with respect to the restoring temperatures T_i^* for different fresh water fluxes $f_1 \in [-0.18595, 0.15]$ compared to data from the more complex model *Climber 2*, compare e.g. [8]. This results in a least squares problem where we want to minimize the sum of the squared differences of the flow rate calculated from the box model and the *Climber* model for different values of the freshwater flux f_1.

$$\min_u J(y,u) := \frac{1}{2}\sum_{i=1}^{N}(m(f_{1,i}) - m_{Climber,i})^2 \quad \text{s.t. } \dot{y} = F(y(f_{1,i}), u) = 0 \;\forall i. \tag{1}$$

The dependence of J on the state vector y appears due to the modeling of the overturning $m(f_{1,i}) = m(y, f_{1,i})$ in *(Box)*.
One straightforward method to optimize the parameters in u is to apply standard optimization algorithms supplied by numerical libraries. Here we used a Quasi-Newton implementation with BFGS update of the Hessian and a simple gradient method. Both methods are supplied in MATLAB's optimization toolbox in the function `fminunc` with the option setting `'LargeScale'='off'` (for both) and `'HessUpdate'='SteepDesc'` (additionally for the gradient method, the BFGS update is the default when this option is not set).
For each f_i, $i = 1, ..., N$ and each parameter value u during the optimization process the box model has to be run into a steady state which takes about 400 (for small f_1) to 1400 (for larger f_1) explicit Euler time steps.
For starting values $u_0 = [0, 0, 0]$, and $N = 68$ the optimization took

- 12 iterations needing 14 pairs of function (which means a numerically converged Euler iteration) plus derivative evaluations (which can be a finite difference or an AD-generated derivative) with the Quasi-Newton method to reduce the scaled cost (i.e. J/N scaled with the number of data points N) to a value of 0.206 and

- 29 iterations with 64 pairs of function plus derivative evaluations with the gradient method to reduce J/N to the value of 0.208,

compare Figure 2. Curves of the fitted overturning m can be seen in Figure 3.
The used line search procedure in both algorithms is the same and only takes *pairs* of function and derivative computations. Thus it makes sense to compare the total number of these pairs of function and derivative computations times the minimum number of Euler steps (needed for the pseudo time-stepping to converge) as a measure of the effort of the two methods, in order to compare them to the One-shot approach later on. Using this measure, it makes no difference whether we use an AD-generated derivative (in forward mode) or one numerically approximated by finite differences. In both cases, one such pair of function and derivative computation will take $n+1$ pseudo time-steppings, if n denotes the number of parameters to be optimized ($n = 3$ in our case). For a higher number n, the reverse mode of AD would be preferable.
In our example we obtained as lower bound:

$$\left.\begin{array}{c}\text{(\#function + derivative}\\ \text{evaluations}\\ \text{in the optimization)}\\ *\\ \text{(\#Euler steps per}\\ \text{function evaluation)}\end{array}\right\} \geq \left\{\begin{array}{rcll} 14\cdot 400 &=& 5600 & \text{(BFGS)}\\ 64\cdot 400 &=& 25600 & \text{(Gradient)}\end{array}\right\} \tag{2}$$

The total number of Euler steps has thus to be multiplied by $(n+1)$, if one is interested in this quantity.

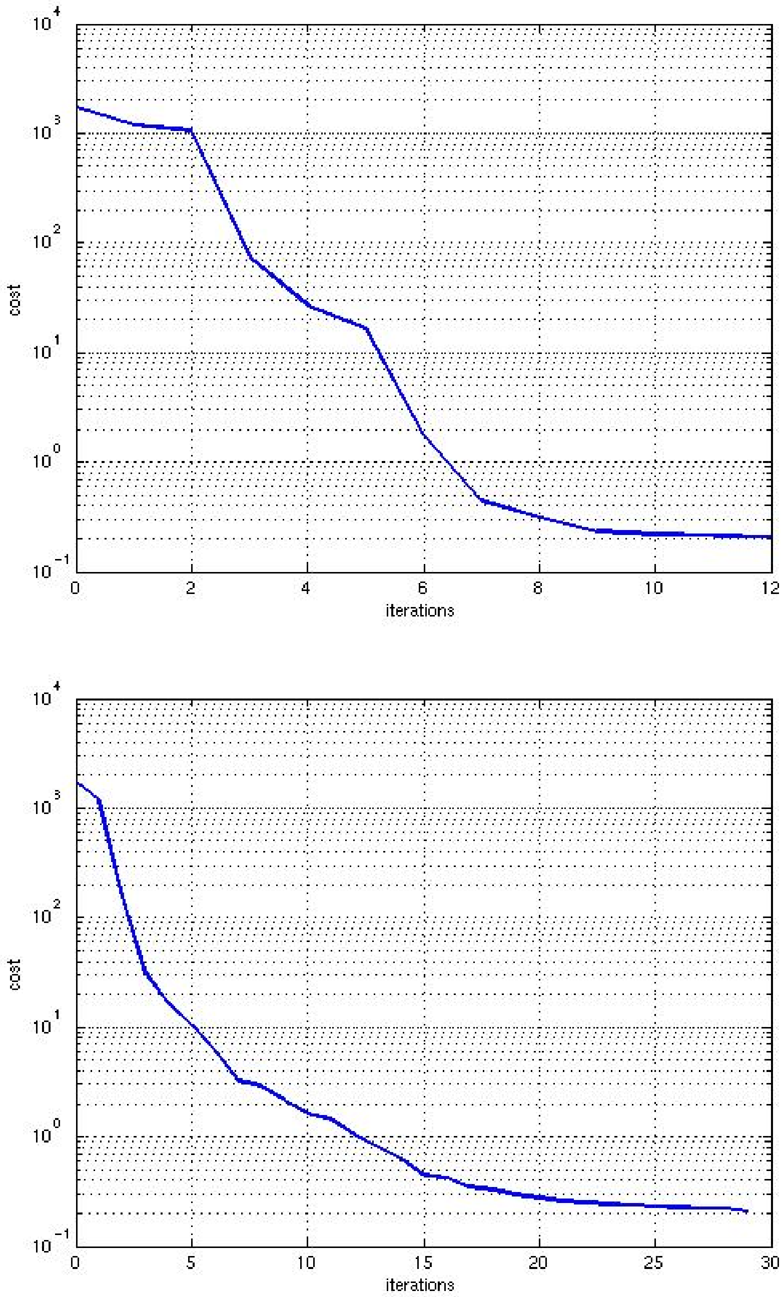

Figure 2: Reduction of the objective during optimization with MATLAB's Quasi-Newton method with BFGS update (top) and gradient method (bottom).

Our aim was to compare the so-called One-shot method by A. Griewank and A. Hamdi [5] to these results. The main idea of the method is to reach optimality while reaching feasibility, which means in our case while running the model into a steady state. Therefore one may hope for a reasonable saving in the total number of Euler steps needed for the optimization.

4 One-shot optimization method

In this section we want to shortly describe the One-shot method introduced in [5]. To avoid confusion with the freshwater flux f_1 we denote the cost by J instead of f as in [5]. We consider the optimization problem in general

$$(P) \quad \min_{y,u} J(y,u) \quad s.t. \quad c(y,u) = 0,$$

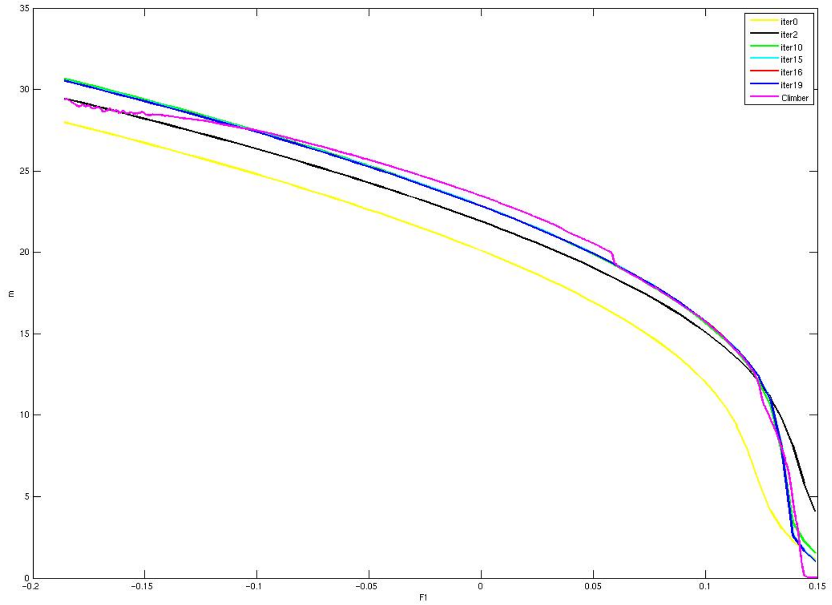

Figure 3: Curves of the fitted overturning $m = m(f_1)$ in different iterations during optimization with MATLAB's Quasi-Newton method with BFGS update.

where $J : Y \times U \to \mathbb{R}$ is the objective function, $u \in U$ the design or parameter vector to be optimized, given in a finite dimensional Hilbert space with $dim(U) = m$. Moreover, $y \in Y$ the state function where Y is a Hilbert space and $c : Y \times U \to Y$ is the state equation. We assume J to be $C^{2,1}$.
The idea is to reformulate the condition $c(y, u) = 0$ into a fixed point equation $y = G(y, u)$ where the iteration function $G : Y \times U \to Y$ is assumed to be $C^{2,1}$ with the contraction factor $\rho < 1$, i.e.

$$\|G_y(y, u)\| = \|G_y^\top(y, u)\| \leq \rho < 1 \quad \Rightarrow \quad \|G(y_1, u) - G(y_2, u)\| \leq \rho \|y_1 - y_2\|.$$

Here and from now on the subscripts denote the partial derivatives w.r.t y and u, respectively. In the finite dimensional case, the associated Lagrangian to problem (P) with the adjoint $\bar{y} \in Y$ is

$$L(y, \bar{y}, u) = J(y, u) + \bar{y}^\top (G(y, u) - y).$$

$(y_*, \bar{y}_*, u_*)$ fulfills the first order necessary optimality condition, if it holds

$$\begin{aligned} 0 = \frac{\partial L}{\partial y} &= J_y(y^*, u^*) + \bar{y}^{*\top} G_y(y^*, u^*) - \bar{y}^{*\top}, \\ 0 = \frac{\partial L}{\partial \bar{y}} &= G(y^*, u^*) - y^*, \\ 0 = \frac{\partial L}{\partial u} &= J_u(y^*, u^*) + \bar{y}^{*\top} G_u(y^*, u^*). \end{aligned}$$

Herewith, the following iteration is reasonable:

$$\begin{array}{rcll} y_{k+1} & = & G(y_k, u) & \text{to reach primal feasibility,} \\ \bar{y}_{k+1}^T & = & \bar{y}_k^\top G_y(y_k, u_k) + J_y(y_k, u_k) & \text{to reach dual feasibility, and} \\ u_{k+1} & = & u_k - B_k^{-1} \left(J_u(y_k, u_k) + \bar{y}_k^\top G_u(y_k, u_k) \right) & \text{to obtain optimality.} \end{array}$$

The update in u resembles a Quasi-Newton step. However, the matrix B must not only provide descent of the Lagrangian, but also convergence of the combination of the three updates. The authors of [5] suggest to choose the preconditioner B as the approximation of the Hessian of an doubly augmented Lagrangian L^a multiplied by a factor σ, where

$$L^a(y, \bar{y}, u) = \frac{\alpha_L}{2} \|G(y, u) - y\|^2 + \frac{\beta_L}{2} \|N_y(y, \bar{y}, u)^\top - \bar{y}\|^2 + N(y, \bar{y}, u) - \bar{y}^\top y,$$

where $N(y, \bar{y}, u) = J(y, u) + \bar{y}^\top G(y, u)$, i.e. $L(y, \bar{y}, u) = N(y, \bar{y}, u) - \bar{y}^\top y$, and $\alpha_L > 0$ and $\beta_L > 0$. The Hessian of L^a is approximately

$$\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu} \approx \nabla_{uu}^2 L^a.$$

Hamdi and Griewank prove in [5] that stationary points of problem (P) are also stationary points of L^a. Furthermore, for the latter descent is provided if

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu})$$

where

$$\sigma := 1 - \rho - \frac{(1 + \frac{\|N_{yy}\|}{2}\beta)^2}{\alpha_L \beta_L (1-\rho)}.$$

Since the calculation of second derivatives is very expensive concerning calculation costs and time, in practice, a Low-Rank update, e.g. BFGS update, is used. It is proposed to choose α_L and β_L as

$$(\alpha_L, \beta_L) = \min_{\alpha,\beta} \frac{\alpha \|G_u\|_2^2 + \beta \|N_{yu}\|_2^2}{\sigma}.$$

Under the assumption that $\sqrt{\alpha_L \beta_L}(1-\rho) > 1 + \frac{\beta_L}{2}\|N_{yy}\|$ holds and $\|N_{yy}\| \neq 0$ we obtain

$$\beta_L = \frac{3}{\sqrt{\|N_{yy}\|^2 + 3\frac{\|N_{yu}\|_2^2}{\|G_u\|_2^2}(1-\rho)^2} + \frac{\|N_{yy}\|}{2}} \text{ and } \alpha_L = \frac{\|N_{yu}\|_2^2 \beta_L (1 + \frac{\|N_{yy}\|}{2}\beta_L)}{\|G_u\|_2^2 (1 - \frac{\|N_{yy}\|}{2}\beta_L)}.$$

Griewank and Hamdi make three major assumptions which are the basis of the convergence of the One-shot method:

1. J is a $C^{2,1}$ function,
2. $\partial c / \partial y$ is always invertible such that for given u there exists only one y with $c(y, u) = 0$ and of course
3. the function $G \in C^{2,1}$ is a fixed point iteration with the contraction factor $\rho < 1$.

5 One-shot method for the advanced version of the 4-box model

In this section, we present the realization of the One-shot method for the advanced box model *(Box)*. Moreover we show how the required derivative code was generated using the software tool *ADiMat*, see [1], for Algorithmic (or Automatic) differentiation (AD). Finally we explicitly compute the used preconditioner B.

5.1 Definitions

Usually, in least squares optimization problems in climate modeling it is assumed that the starting value u_0 is not too far away from the solution u_* and a so called correction term $\frac{\alpha_{corr}}{2}\|u - u_0\|^2$ is added to the objective function. In the 4-box model we thus add the term

$$\frac{\alpha_{corr}}{2} \sum_{i=1}^{3} (T_i^* - T_{i,0}^*)^2$$

to the objective J in (1).

The iteration function G is in our testings the explicit Euler time stepping. It can be proven that the explicit Euler time stepping is a fixed point iteration and converges for step size $\Delta t = 1$ and parameters of the box model chosen. For a sketch of the proof we refer to the Appendix.

We remark that here G is not a function from $\mathbb{R}^8$ to $\mathbb{R}^8$, but from $\mathbb{R}^{8N}$ to $\mathbb{R}^{8N}$, because the update has to be done simultaneously for all freshwater fluxes $f_{1,i}$, $i = 1, ..., N$. The Jacobian G_y then is a diagonal block matrix with entries $G_{y(f_{1,i})}$, $i = 1, ...N$. Analogously the cost is a function $J : \mathbb{R}^{8N} \times \mathbb{R}^3 \to \mathbb{R}$. In our case J as a least squares functional clearly satisfies the regularity assumptions. For the assumed regularity of G it was crucial to use the advanced or smoothed version of the box model *(Box)* which eliminates the kink at states where $m = 0$.

5.2 Calculation of derivatives using Automatic Differentiation

For the update of the adjoint $\bar{y}$ we calculated the Jacobian G_y and the gradient J_y with the automatic differentiation tool *ADiMat* which uses the source code transformation approach to augment MATLAB codes, see `www.sc.rwth-aachen.de/vehreschild/adimat` and e.g. [1]. With the help of a MATLAB script file `driver.m` we developed the derived code. In the file we had to call the function to be differentiated and declare the independent and dependent variable and the top routine. In our case `driver.m` contained:

```
ynew = G(y, u, F1)

% ADiMat AD_IVARS = y
% ADiMat AD_DVARS = ynew
% ADiMat AD_TOP = G
```

Calling `adimat driver.m` in the shell produces the files `g_G.m` and `g_boxmodel.m`. `g_G.m` contains

```
function [g_eulerstep, eulerstep] = g_G(g_y, y, u, F1)

dt= 1.0;

[g_tmp_boxmodell_0000, tmp_boxmodell_0000]= g_boxmodell(g_y, y, u, F1);
g_tmp_G_0000= dt* g_tmp_boxmodell_0000;
tmp_G_0000= dt* tmp_boxmodell_0000;
g_eulerstep= g_y+ g_tmp_G_0000;
eulerstep= y+ tmp_G_0000;
%endclear tmp_boxmodell_0000 tmp_G_0000 g_tmp_boxmodell_0000 g_tmp_G_0000 ;
```

Here, `eulerstep` is the original output vector called `ynew` in `driver.m` and `g_eulerstep` is the derivative of `eulerstep` with respect to `y` multiplied by the seed matrix `g_y` which is the identity matrix for the computation of the full Jacobian. *ADiMat* provides derivatives using the forward mode of automatic differentiation. As described in [3] costs of the calculation of the Jacobian add up to a relatively small multiple of n times the costs for the evaluation of the function itself where n is the dimension of the variable with respect to which to be differentiated. In our case, we have $n = 8$ that means costs for the calculation of the Jacobian remain acceptably small.
Below is just a small extract of file `g_boxmodel.m`:

```
function [g_F, F]= g_boxmodell(g_y, y, u, F1)
⋮
g_mminus= ls_mquot(-g_m, -m, g_nenner2, nenner2);
mminus= -m/ nenner2;
g_tmp_boxmodell_00008= lam1* ((-g_y(1)));
tmp_boxmodell_00008= lam1* (Tr1+ W1- y(1));
g_tmp_boxmodell_00009= g_mplus/ V1;
tmp_boxmodell_00009= mplus/ V1;
g_tmp_boxmodell_00010= g_y(4)- g_y(1);
tmp_boxmodell_00010= y(4)- y(1);
g_tmp_boxmodell_00011= ls_mprod(g_tmp_boxmodell_00009,tmp_boxmodell_00009,
(g_tmp_boxmodell_00010), (tmp_boxmodell_00010));
tmp_boxmodell_00011= tmp_boxmodell_00009*(tmp_boxmodell_00010);
g_tmp_boxmodell_00012= g_mminus/ V1;
```

```
tmp_boxmodell_00012= mminus/ V1;
g_tmp_boxmodell_00013= g_y(3)- g_y(1);
tmp_boxmodell_00013= y(3)- y(1);
g_tmp_boxmodell_00014= ls_mprod(g_tmp_boxmodell_00012,tmp_boxmodell_00012,
(g_tmp_boxmodell_00013), (tmp_boxmodell_00013));
tmp_boxmodell_00014= tmp_boxmodell_00012* (tmp_boxmodell_00013);
g_F(1)= g_tmp_boxmodell_00008+ g_tmp_boxmodell_00011+g_tmp_boxmodell_00014;
F(1)= tmp_boxmodell_00008+ tmp_boxmodell_00011+tmp_boxmodell_00014;
clear tmp_boxmodell_00008 tmp_boxmodell_00009 tmp_boxmodell_00010 ...
```

We noticed that tests with codes provided by *ADiMat* needed a lot of computational time. In our opinion, this is mainly caused by the introduction of a Matlab class for the seed matrix in *ADiMat*. Since our code is not too complex, we transformed the *ADiMat* output into a Matlab code with a simple matrix instead of an object for the seed matrix. We then obtained significant time savings. Surely, for more complex models this transformation is not that easy to handle.

5.3 The preconditioner B

As proposed in [5] and to choose an approximation to the Hessian of the doubly augmented Lagrangian, we have to consider its components G_u, N_{yu} and N_{uu}. Here the special structure of the cost and the model equations can be exploited. We obtain

$$\begin{aligned} G_u^\top G_u &= \Delta t^2 \frac{\partial F}{\partial u}^\top \frac{\partial F}{\partial u} = \Delta t^2 \begin{pmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{pmatrix} \\ N_{uu} &= J_{uu} + \bar{y}^\top G_{uu} = \alpha_{corr} I_3 + 0 \end{aligned}$$

where I_3 is the identity in $\mathbb{R}^{3\times 3}$. Since J_{yu} and G_{yu} are both zero, also N_{yu} is zero. We obtain for $\Delta t = 1$ that

$$\begin{aligned} L_{uu}^a &= \begin{pmatrix} \alpha_{corr} + \alpha_L \lambda_1^2 & 0 & 0 \\ 0 & \alpha_{corr} + \alpha_L \lambda_2^2 & 0 \\ 0 & 0 & \alpha_{corr} + \alpha_L \lambda_3^2 \end{pmatrix} \text{ and} \\ (L_{uu}^a)^{-1} &= \begin{pmatrix} \frac{\alpha_L^{-1}}{\alpha_L^{-1}\alpha_{corr}+\lambda_1^2} & 0 & 0 \\ 0 & \frac{\alpha_L^{-1}}{\alpha_L^{-1}\alpha_{corr}+\lambda_2^2} & 0 \\ 0 & 0 & \frac{\alpha_L^{-1}}{\alpha_L^{-1}\alpha_{corr}+\lambda_3^2} \end{pmatrix}. \end{aligned}$$

Obviously, L_{uu} is constant which means also the preconditioner B remains constant during the iteration process, except that the factor σ changes, because of the change in $\|N_{yy}\|$.
We do not use the exact values of σ but inexact estimations assuming that σ decreases during the optimization process, and such enlarging the factor $1/\sigma$ every 50 iterations by the factor 5, where those values result from different tests which showed that the performance was best with this choice. The values α_L and β_L could not be computed the way Hamdi and Griewank propose in [5], because $\|N_{yu}\| = 0$ and therewith also $\alpha_L = 0$ which contradicts the condition $\sqrt{\alpha_L \beta_L}(1-\rho) > 1 + \frac{\beta_L}{2}\|N_{yy}\|$. Using this condition one can estimate that α_L and β_L must be large because $1-\rho$ is quite small for the explicit Euler time stepping.
For the one shot update only α_L is of importance. We chose a very large $\alpha_L = 1000$, because the values $\|G(y,u) - y\|$ are extremely small compared to the values of $J(y,u)$. On the contrary, the correction factor α_{corr} in the objective function is kept small, i.e. $\alpha_{corr,0} = 10$ and is even decreased during the iteration, such that importance is put on the minimization of the differences of the flow rates calculated by the box model and data from the Climber model.

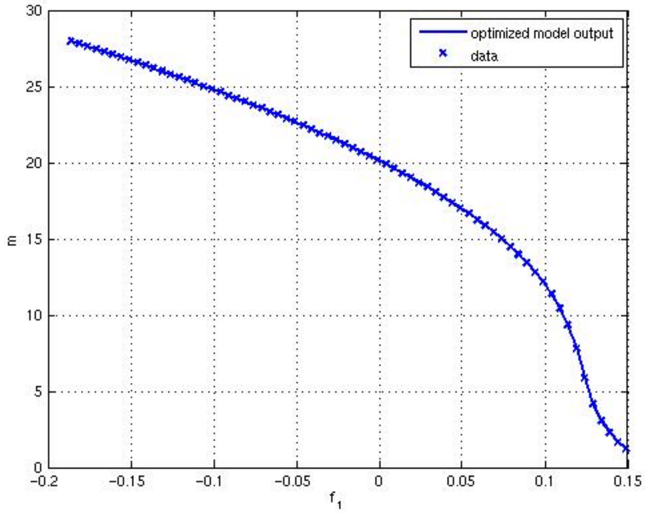

Figure 4: Model output optimized by the One-shot method fit to ideal data.

6 Numerical results

In this section we present numerical results of the optimization using the One-shot approach. We first generated ideal data by the model itself that we used instead of $m_{Climber}$ in (1). Then we optimized to fit the *Climber* data set.

6.1 Fitting model-generated, ideal data

The first test was to fit values of the box model to ideal data which were calculated by running the box model *(Box)* with values T_i^* taken from [11] into a steady state applying the explicit Euler time stepping. The advantage of this data set is that we have a relatively smooth curve without zig-zags as we have in the *Climber* data. That implies that the minimum value which should be found by the optimization procedure is $J = 0$.

Stopping criteria were firstly, that the norm of the gradient of the augmented Lagrangian was less than $\varepsilon = 10^{-5}$ or secondly, there was nearly no change in the norm of $y_{new} - y$, i.e. $\|G(y,u) - y\| < \varepsilon/10^3$ or thirdly, there was nearly no change in the norm of $u_{new} - u$, i.e. $\|u_{new} - u\| < \varepsilon/10^5$ or fourthly, there was no change in the augmented Lagrangian, i.e. $\|L^a_{new} - L^a\| < \varepsilon/10^3$. The One-shot procedure found the correct solution within 2750 iterations. The calculated data nearly completely fit the provided data as can be seen in Figure 4. For larger f_1 the values were not fitted exactly which might be caused by the fact that for larger f_1 the explicit Euler time stepping needs even without optimization many iterations to reach a steady state. Weighted by the number of different f_1, which was in our testing $N = 68$, the difference between the data was reduced from approximately 54 to 0.00001, see Figure 5. The optimization stopped because there was nearly no change in the augmented Lagrangian anymore.

6.2 Fitting the *Climber* data

In the second test, we fit calculated values of the flow rate m to data from the *Climber* model which were also used in [9] and [11]. We did not expect to achieve a curve that totally fits the *Climber* data, and thus not a minimum value of $J = 0$. Here too, the One-shot method found an acceptable solution, see Figure 7. The scaled cost (J/N, again N denoting the number of data points) was reduced to a value of 0.18 within 2175 iterations, see Figure 6. Again the optimization stopped because there was nearly no change in the augmented Lagrangian anymore.

To investigate the numerical effort we compared the number of steps in the One-shot optimization

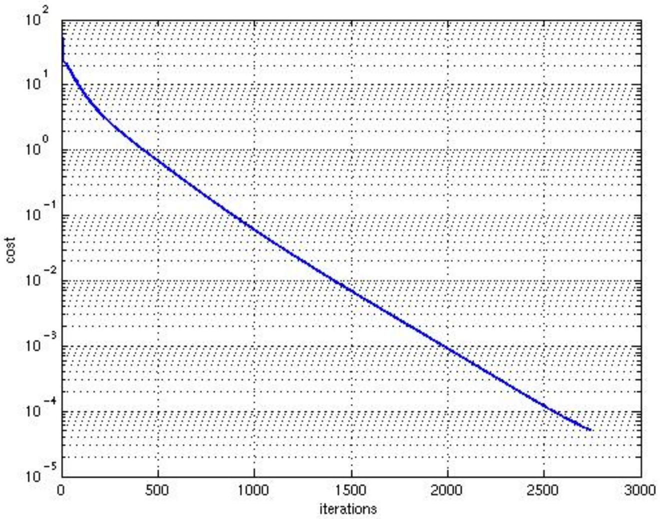

Figure 5: Reduction in the objective function achieved by the One-shot method in the fit to ideal data.

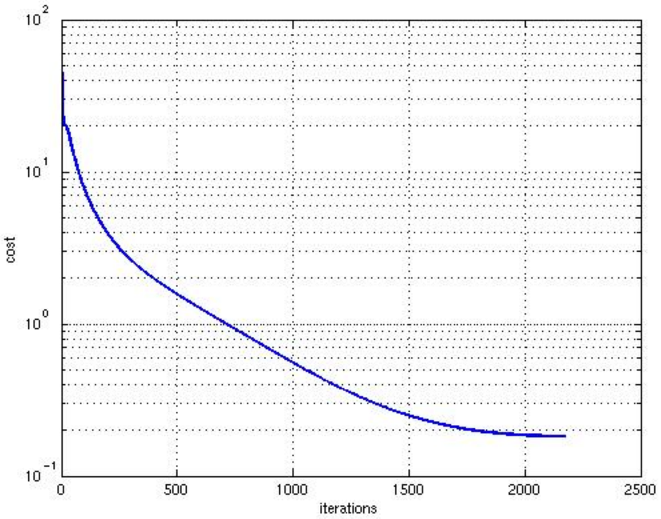

Figure 6: Reduction in the objective function achieved by the One-shot method in the fit to data from *Climber*.

(each step can be seen as a single Euler step for the coupled system of state, adjoint, and parameter) to the number of pairs of function + derivative computations times Euler steps for the Quasi-Newton and gradient optimizers, respectively. Thus we have in this test case a relation of

$$\begin{aligned} \text{Quasi-Newton method} &: \text{One-shot approach} = 5600 : 2175 \approx 2.6 : 1 \\ \text{Gradient method} &: \text{One-shot approach} = 25600 : 2175 \approx 11.8 : 1. \end{aligned}$$

7 Preliminary Summary and Outlook

The One-shot method proved to be a useful tool to find a solution to least squares problems where the goal was to optimize parameters of the advanced box model (*Box*). The obtained numerical optimum

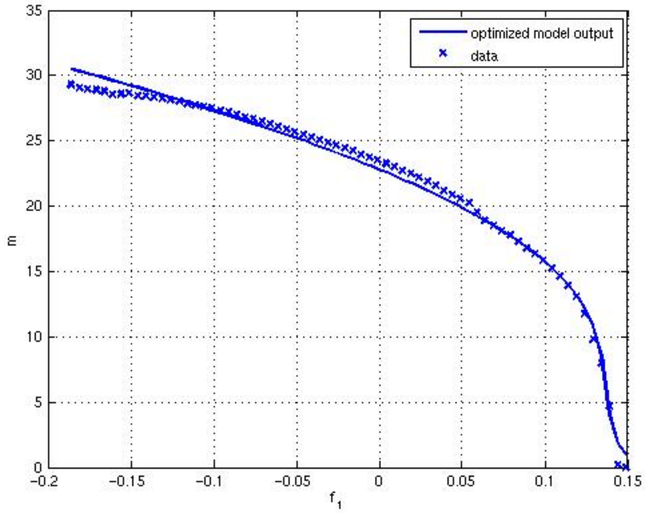

Figure 7: Model output optimized by the One-shot method fit to data from *Climber*.

for the One-shot method was slightly better, moreover the total number of needed time steps was reduced significantly. The results are summarized in Table 1 and Figure 8.

Method	final cost function value (scaled by number of data points)	# optimization steps	# function + derivative evaluations	# Euler steps (*lower bound)
Quasi-Newton	0.206289	12	14	5600*
Gradient	0.207861	29	64	25600*
One-shot	0.182514	2175	1	2175

Table 1: Comparison of optimization results and effort for the three tested optimization methods. Important numbers for the runtime are those in the last column. For the total numbers of Euler steps the values of the last column have to be multiplied by $(n+1)$ when using finite differences or forward mode AD.

Since in our tests those parameters u occur only linearly in the model description, derivatives with respect to u become simple. However, as we have applied an AD tool for the calculation of derivatives with respect to y, that tool could also be used to calculate derivatives with respect to u and therefore, also for other parameters the One-shot method could be a good optimization tool. That will be analyzed in future tests. Of course, improvements could be made by applying a line-search procedure or by calculating better values for $\|N_{yy}\|$ such that the factor $1/\sigma$ of the preconditioner B corresponds to the theory done in [5].

One future goal then is to apply the One-shot method on different parameters of the box model, and more complex equations in ocean or climate modeling.

Appendix: Contraction property of the explicit Euler scheme for the box model

We want to prove that for the advanced box model *(Box)*, the explicit Euler time stepping is a contraction. Therefore, we show that $\|G_y\| < 1$. We have $G_y(y,u) = I + \Delta t F_y(y,u)$. We choose the $\|\cdot\|_\infty$ norm. Here, we analyze in detail only the first line. We compute

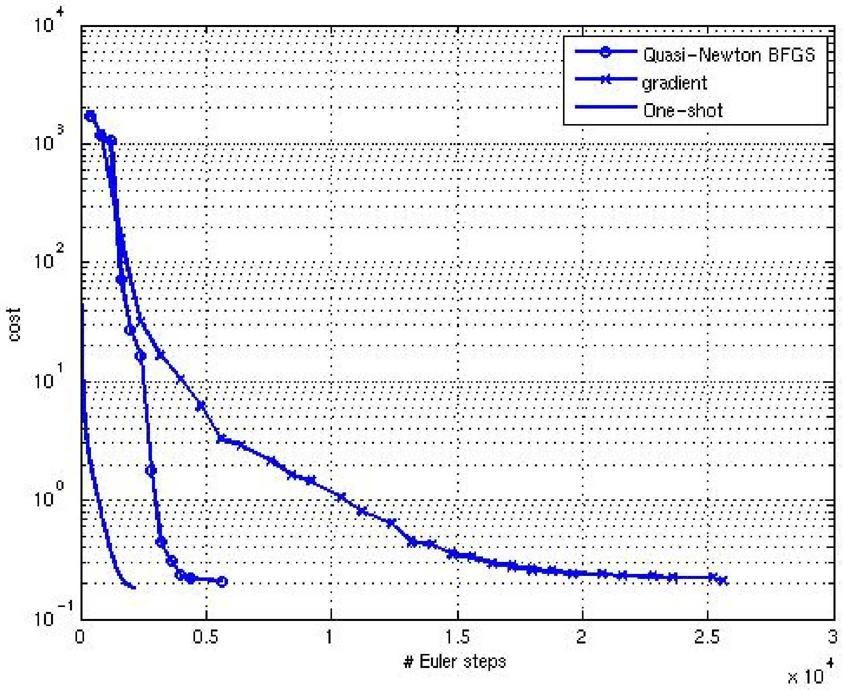

Figure 8: Comparison of cost function reduction w.r.t. to number of Euler steps for the three tested optimization methods. W.r.t. the Euler steps, the optimization by the One-Shot method is nearly done when the other two methods have finished their first optimization steps.

$$
\begin{aligned}
&\sum_{j=1}^{8} |G_y(y,u)_{1,j}| \;= \\
= \;& |1+\Delta t\left(-\lambda_1 + \frac{1}{V_1}(-(m^+ + m^-) + \frac{\alpha k(1-e^{-am}) - \alpha kame^{-am}}{(1-e^{-am})^2}(T_4 - T_1)\right. \\
& \left. + \frac{-\alpha k(1-e^{am}) - \alpha kame^{am}}{(1-e^{am})^2}(T_3 - T_1))\right)| \\
& + \frac{\Delta t}{V_1}|\frac{-\alpha k(1-e^{-am}) + \alpha kame^{-am}}{(1-e^{-am})^2}(T_4 - T_1) + \frac{\alpha k(1-e^{am}) + \alpha kame^{am}}{(1-e^{am})^2}(T_3 - T_1)| \\
& + \frac{\Delta t}{V_1}(|m^-| + |m^+|) \\
& + \frac{\Delta t}{V_1}|\frac{-\beta k(1-e^{-am}) + \beta kame^{-am}}{(1-e^{-am})^2}(T_4 - T_1) + \frac{\beta k(1-e^{am}) + \beta kame^{am}}{(1-e^{am})^2}(T_3 - T_1)| \\
& + \frac{\Delta t}{V_1}|\frac{\beta k(1-e^{-am}) - \beta kame^{-am}}{(1-e^{-am})^2}(T_4 - T_1) + \frac{-\beta k(1-e^{am}) - \beta kame^{am}}{(1-e^{am})^2}(T_3 - T_1)| \\
= \;& |1+\frac{\Delta t}{V_1}\left(-V_1\lambda_1 + -(m^+ + m^-) + \frac{\alpha k(1-am^+e^{-am})}{1-e^{-am}}(T_4 - T_1)\right. \\
& \left. - \frac{\alpha k(1-am^-e^{am})}{1-e^{am}}(T_3 - T_1)\right)| + \frac{\Delta t}{V_1}(|m^-| + |m^+|) \\
& + \frac{\Delta t}{V_1}| - \frac{\beta k(1-am^+e^{-am})}{1-e^{-am}}(T_4 - T_1) + \frac{\beta k(1-am^-e^{am})}{1-e^{am}}(T_3 - T_1)| \\
& + \frac{\Delta t}{V_1}|\frac{\beta k(1-am^+e^{-am})}{1-e^{-am}}(T_4 - T_1) - \frac{\beta k(1-am^-e^{am})}{1-e^{am}}(T_3 - T_1)|
\end{aligned}
$$

Under the condition that

$$|1 - \Delta t(-\lambda_1 + \frac{1}{V_1}(-(m^+ + m^-))| \geq 0,$$

which is the case if

$$\Delta t \leq 1/(\lambda_1 + V_1^{-1}(m^+ + m^-)),$$

and since the fractions in front of $T_4 - T_1$ and $T_3 - T_1$, respectively, are positive, we obtain the inequality

$$\begin{aligned} \sum_{j=1}^{8} |G_y(y,u)_{1,j}| \quad \leq \quad & |1 + \frac{\Delta t}{V_1}(-V_1\lambda_1 + -(m^+ + m^-))| + \frac{\Delta t}{V_1}(|m^-| + |m^+|) \\ & + \frac{\Delta t}{V_1}\frac{2(\alpha+\beta)k(1 - am^+e^{-am})}{1 - e^{-am}}|T_4 - T_1| \\ & + \frac{\Delta t}{V_1}\frac{2(\alpha+\beta)k(1 - am^-e^{am})}{1 - e^{am}}|T_3 - T_1|. \end{aligned}$$

Finally, it holds $1 > \sum_{j=1}^{8} |G_y(y,u)_{1,j}|$ if

$$\begin{aligned} 1 \quad > \quad & 1 - \Delta t\lambda_1 + \frac{\Delta t}{V_1}\frac{2(\alpha+\beta)k(1 - am^+e^{-am})}{1 - e^{-am}}|T_4 - T_1| \\ & + \frac{\Delta t}{V_1}\frac{2(\alpha+\beta)k(1 - am^-e^{am})}{1 - e^{am}}|T_3 - T_1|, \\ \lambda_1 \quad > \quad & \frac{2(\alpha+\beta)k(1 - am^+e^{-am})}{V_1(1 - e^{-am})}|T_4 - T_1| + \frac{2(\alpha+\beta)k(1 - am^-e^{am})}{(V_1(1 - e^{am})}|T_3 - T_1| \end{aligned}$$

We remark that those limitations on Δt and λ_1 are easily to fulfill because the box volume V_1 is very large, i.e. $V_1 >> 1$.
Similar restrictions can be made for all lines of G_y, such that one can choose parameters such that for all $i = 1, ..., 8$ the sum $\sum_{j=1}^{8} |G_y(y,u)_{i,j}|$ is less than one and the explicit Euler time stepping is a contraction.

Acknowledgements

The authors would like to thank Adel Hamdi, Andreas Griewank, Stefan Rahmstorf, Kirsten Zickfeld and Andre Vehreschild for support.

References

[1] Christian H. Bischof, Bruno Lang, and Andre Vehreschild. Automatic differentiation for MATLAB programs. *Proceedings in Applied Mathematics and Mechanics*, 2(1):50–53, 2003.

[2] B. Christianson. Reverse Accumulation and Implicit Functions. *Optimization Methods and Software*, 9(4):307–322, 1998.

[3] Andreas Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.

[4] Andreas Griewank and Adel Hamdi. Properties of an augmented lagrangian for design optimization. *Preprint-Nr.: SPP1253-11-01*, 2008.

[5] Andreas Griewank and Adel Hamdi. Reduced quasi-newton method for simultaneous design and optimization. *Preprint-Nr.: SPP1253-11-02*, 2008.

[6] Thomas Kaminski, Ralf Giering, and Michael Voßbeck. Efficient sensitivities for the spin-up phase. In H. M. Bcker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors, *Automatic Differentiation: Applications, Theory, and Implementations*, pages 283–291. Springer, New York, 2005. volume 50 of Lecture Notes in Computational Science and Engineering.

[7] Emre Özkaya and Nicolas Gauger. Single-step one-shot aerodynamic shape optimization. *Preprint SPP1253-10-04*, 2008.

[8] S. Rahmstorf, V. Brovkin, M. Claussen, and C. Kubatzki. Climber-2: A climate system model of intermediate complexity. part ii: Model sensitivity. *Clim. Dyn.*, 17:735–751, 2001.

[9] Thomas Slawig and Kirsten Zickfeld. Parameter optimization using algorithmic differentiation in a reduced-form model of the atlantic thermohaline circulation. *Nonlinear Analysis: Real World Applications*, 5/3:501–518, 2004.

[10] Sven Titz, Till Kuhlbrodt, Stefan Rahmstorf, and Ulrike Feudel. On freshwater-dependent bifurcations in box models of the interhemispheric thermohaline circulation. *Tellus A*, 54:89 – 98, 2002.

[11] Kirsten Zickfeld, Thomas Slawig, and Stefan Rahmstorf. A low-order model for the response of the atlantic thermohaline circulation to climate change. *Ocean Dynamics*, 54:8–26, 2004.

Oneshot Parameter Identification - Simultaneous Model Spin-up and Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation

Claudia Kratzenstein[*], Thomas Slawig[†]

(Published on the Preprint server of the DFG-SPP 1253 in September 2009; available at: http://www.am.uni-erlangen.de/home/spp1253/wiki/images/7/7c/Preprint-SPP1253-082.pdf)

Abstract

Parameters of a box model of the north Atlantic thermohaline circulation are optimized to fit the model results, i.e. the overturning, to data given by a more detailed climate model of intermediate complexity. Since the model is run into a steady state by a pseudo time-stepping, efficient techniques are necessary to avoid extensive recomputations and/or storing when using adjoint-based gradient representations for local optimization algorithms. The preconditioned Oneshot approach studied by Hamdi and Griewank that simultaneously updates state, adjoint and parameter values, is applied to this nonlinear climate model. For the required partial derivatives, a software tool of Algorithmic/Automatic Differentiation, namely TAF, was used. Numerical results are compared to results obtained by the BFGS-Quasi-Newton method.

Keywords: Parameter optimization, pseudo time-stepping, fixed point solver, climate model, automatic differentiation, bounded retardation

1 Introduction

Parameter optimization is an important task in all kind of climate models or models that simulate parts of the climate systems, as for example ocean or atmospheric models. Still, some processes are not well-known, some are too small-scaled in time or space, and others are just beyond the scope of the model. All these processes are *parameterized*, i.e. simplified model functions (parameterizations) are used. These necessarily include lots of – most of the time – only heuristically known parameters. A main task thus is to *calibrate* the models by optimizing the parameter w.r.t. to data from measurements or other (more complex) models.

Similar to many applications in engineering applications of fluid mechanics, also in geophysical flows (e.g. ocean models) an optimization is at first performed for steady states of the equations before proceeding to transient problems. This means that only the stationary solution is used in the cost or objective function to be minimized. Moreover (and this the second point where engineering and geophysical flow problems are similar), the computation of steady states is often performed by running a transient model into the steady state. This strategy is called pseudo time-stepping, since the time variable may be regarded as a kind of iteration counter.

It is well known from optimal control of differential equations that the classical adjoint technique (that allows the representation of the gradient of the cost) leads to a huge amount of recomputations, storing or both. This problem looks even more annoying in the pseudo-time stepping context, since here only the final, numerically converged state is important for the cost. Nevertheless a classical adjoint technique would need all intermediate iterates.

[*] ctu@informatik.uni-kiel.de, Institut f. Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany. Research supported by DFG SPP 1253 and DFG Cluster of Excellence *The Future Ocean*.

[†] ts@informatik.uni-kiel.de, Institut f. Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany. Research supported by DFG SPP 1253 and DFG Cluster of Excellence *The Future Ocean*.

If the number of parameters to be optimized is small, a sensitivity equation approach is also reasonable. On the discrete level this is comparable to the application of the forward mode of Automatic or Algorithmic Differentiation (AD). Here, the sensitivity equation has the same temporal integration direction (namely forward) as the original pseudo time-stepping. But nevertheless it is worth while investigating how the two (for a non-linear model) coupled iterations for state and sensitivity are performed.

Griewank described in [4] the differences between two-phase (where the iteration for the state is run to the steady state or fixpoint first, and then the sensitivity is computed) and piggy-back approaches (where both iterations are combined to one). Christianson in [2] proposed to perform the sensitivity iteration with the converged state instead of using its iterates. Giering, Kaminski and Vossbeck in [7] used the so-called *Full Jacobian approach*, where they directly used the steady state equation and differentiated it to obtain an equation for the gradient.

The approach used here is called Oneshot approach, was developed by Griewank and Hamdi, and can be seen as an extension of the piggy-back strategy aiming for optimality and feasibility simultaneously with so-called bounded retardation. That means that the number of Oneshot iterations shall not too much exceed the number of fixed point iterations necessary for the model spin-up itself. The theoretical results were published in [6],[5], an engineering application was presented by Özkaya and Gauger in [9].

In this paper, we apply the Oneshot approach to a box model of the North Atlantic. The authors have analysed the Oneshot approach for this ocean model already in [8], but with only three optimization parameters and comparatively simple derivatives due to the linear appearance of the parameters in the model. We want to minimize the differences between the output of the steady state of the model compared to data given by a more complex coupled climate model. The box model is a non-linear ODE system with eight equations for temperatures and salinities in four ocean compartments. As parameters to be optimized we choose here altogether six model parameters which include the so-called restoring temperatures, which can be seen as counterparts of the three surface temperatures, two coupling constants, and a parameter influencing the deviation from the physically correct model. Here, the parameters enter in an nonlinear fashion resulting in so-called non-separable adjoints where the adjoint is no longer the sum of a term on the state and a term on design.

The outline of this paper is the following. In the next section we describe the box model in its original version and in a modified, differentiable version that we used here. In Section 3 we present the formulation of the optimization problem. In the fourth section we briefly review the Oneshot method, and analyse its theoretical basis, namely the assumptions for convergence, in the next section in the particular case of the box model. In Section 6 we present details of the implementation and numerical results of the Oneshot method comparing the latter with results obtained by the BFGS-Quasi-Newton approach. Finally, in Section 7, we give some concluding remarks and an outlook to future work.

2 The Box Model of the Thermohaline Circulation

In this section we present the formulation of the climate model, which we examine, a so-called box model. We start with the original version by S. Rahmstorf et. al. [13] and proceed with an advanced version which provides a differentiable right-hand side of the ODE system. Moreover, we describe the applied time-stepping scheme.

2.1 Introduction to the 4-Box Model

The basic 4-box model of the Atlantic Thermohaline Circulation described by S. Rahmstorf, K. Zickfeld and T. Slawig [13] simulates the flow rate of the Atlantic Ocean known as the 'conveyor belt', carrying heat northward and having a significant impact on climate in northwestern Europe. Temperatures T_i and salinity differences S_i in four different boxes $i = 1, ..., 4$, the southern, northern, tropical and the deep Atlantic, are the characteristics inducing the flow rate. The surface boxes exchange heat and freshwater with the overlying atmosphere, which causes a pressure-driven circulation. The following nonlinear system of ordinary differential equations describes the flow for the present circulation direction.

$$\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m}{V_1}(T_4 - T_1) & \dot{S}_1 &= \tfrac{S_0 f_1}{V_1} + \tfrac{m}{V_1}(S_4 - S_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m}{V_2}(T_3 - T_2) & \dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m}{V_2}(S_3 - S_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m}{V_3}(T_1 - T_3) & \dot{S}_3 &= \tfrac{S_0(f_2 - f_1)}{V_3} + \tfrac{m}{V_3}(S_1 - S_3)\\
\dot{T}_4 &= \tfrac{m}{V_4}(T_2 - T_4) & \dot{S}_4 &= \tfrac{m}{V_4}(S_2 - S_4)\\
m &= k(\beta(S_2 - S_1) - \alpha(T_2 - T_1));
\end{aligned}$$

where m is the meridional volume transport or overturning. The constants f_1, f_2 are freshwater fluxes containing atmospheric water vapor transport and wind-driven oceanic transport. They are multiplied by a reference salinity S_0 for conversion to a salt flux. Moreover, k is a hydraulic constant linking volume transport m to the density difference, α, β are expansion coefficients and T_i^* temperatures towards which the surface boxes $i = 1, ..., 3$ are relaxed. λ_i are individual coupling constants computed from the thermal coupling constant Γ and the respective box thickness. V_i are the volumes of boxes $i = 1, ..., 4$. Figure 1 illustrates the flow.

In [13] the authors apply the Explicit Euler time stepping to run the model into a steady state. m is set to zero if a negative flow rate is calculated. That means this model simulates only the flow with an upwelling in the south and downwelling in the north, which is the present circulation direction. For the inverse direction the advective terms of the model must be reformulated.

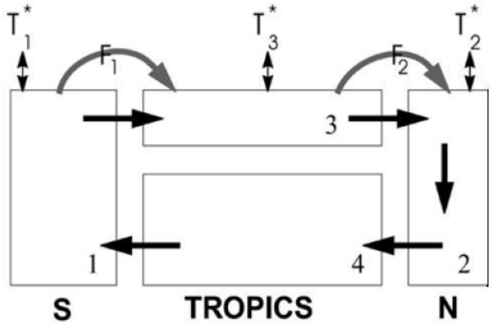

Figure 1: The Rahmstorf 4-box model

The disadvantage of the model is that it is not differentiable for $m = 0$, but the theory of the Oneshot approach used here requires a differentiable model.

2.2 The Differentiable Version of the Box Model

Titz, Kuhlbrodt, Rahmstorf and Feudel [12] propose a coupling of the two flow directions to obtain a smoothed version of the box model. The ODEs of the resulting advanced box model look as follows:

$$\begin{aligned}
\dot{T}_1 &= \lambda_1(T_1^* - T_1) + \tfrac{m^+}{V_1}(T_4 - T_1) + \tfrac{m^-}{V_1}(T_3 - T_1)\\
\dot{T}_2 &= \lambda_2(T_2^* - T_2) + \tfrac{m^+}{V_2}(T_3 - T_2) + \tfrac{m^-}{V_2}(T_4 - T_2)\\
\dot{T}_3 &= \lambda_3(T_3^* - T_3) + \tfrac{m^+}{V_3}(T_1 - T_3) + \tfrac{m^-}{V_3}(T_2 - T_4)\\
\dot{T}_4 &= \tfrac{m^+}{V_4}(T_2 - T_4) + \tfrac{m^-}{V_4}(T_1 - T_4)\\
\dot{S}_1 &= \tfrac{S_0 f_1}{V1} + \tfrac{m^+}{V_1}(S_4 - S_1) + \tfrac{m^-}{V_1}(S_3 - S_1)\\
\dot{S}_2 &= -\tfrac{S_0 f_2}{V_2} + \tfrac{m^+}{V_2}(S_3 - S_2) + \tfrac{m^-}{V_2}(S_4 - S_2)\\
\dot{S}_3 &= \tfrac{S_0(f_2 - F_1)}{V_3} + \tfrac{m^+}{V_3}(S_1 - S_3) + \tfrac{m^-}{V_3}(S_2 - S_4)\\
\dot{S}_4 &= \tfrac{m^+}{V_4}(S_2 - S_4) + \tfrac{m^-}{V_4}(S_1 - S_4)
\end{aligned} \qquad (Box)$$

where for some positive a,

$$m^+ = \frac{m}{1-e^{-am}}, \qquad m^- = \frac{-m}{1-e^{am}}, \qquad m = k(\beta(S_2 - S_1) - \alpha(T_2 - T_1)).$$

The deviation from the physically correct model becomes smaller the larger a is, because using the factors m^+ and m^- the last summand nearly vanishes for a clockwise flow, i.e. $m > 0$, and the middle term almost vanishes for the opposite direction, i.e. $m < 0$, as shown in figure 2.

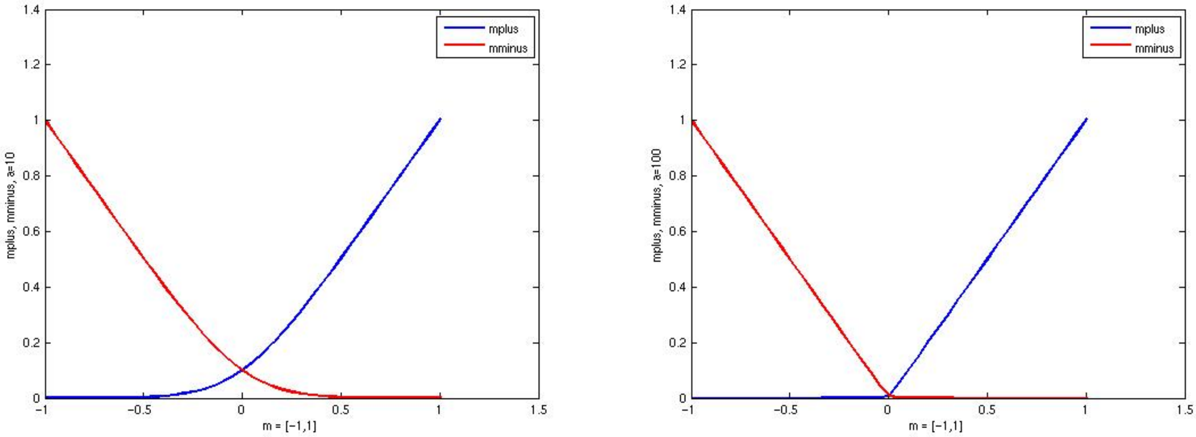

Figure 2: m^+ and m^- for $a = 10$ and $a = 100$ and m close to 0.

3 The Optimization Problem

In this section we formulate the optimization problem.

Our goal is to optimize the smoothed model with respect to the restoring temperatures T_i^*, $i = 1, 2, 3$, the coupling constant k, the thermal coupling constant Γ which appears in the calculation of λ_i and the parameter a. This is done by fitting the obtained overflow m for different fresh water fluxes $f_1 \in [-0.18595, 0.15]$ to data $m_{Climber}$ from the more complex ocean model *Climber 2*, compare e.g. [10]. m has to be multiplied by a factor $\delta = \frac{\text{reference boxvolume}}{10^6 * \text{seconds per year}} \approx 3170.98$ for conversion into the overflow unit in the *Climber* model.

This results in a least squares problem where we minimize the sum of the squared differences of the flow rate calculated from the box model and from the *Climber* model for N different values of the freshwater flux f_1. A penalty term $\frac{\alpha_w}{2}\|u - u_0\|_2^2$, $\alpha_w > 0$, is added to keep the model parameters $u = (T_1^*, T_2^*, T_3^*, \Gamma, k, a)$ near estimated values u_0 which act as the starting value. Otherwise, model parameters might get too far away from real world values.

In the following, we denote the state $y_i = y(f_{1,i}) = (T_1, T_2, T_3, T_4, S_1, S_2, S_3, S_4)_i$, $i = 1, ..., N$, model parameters $u = (T_1^*, T_2^*, T_3^*, \Gamma, k, a)$ as above, and $F(y, u)$ as the right hand side of the system of ODEs of the advanced box model *(Box)*. Hence, the minimization problem is

$$\begin{aligned} \min_{y,u} J(y,u) \quad &:= \quad \frac{1}{2}\|\delta m - m_{Climber}\|_2^2 + \frac{\alpha_w}{2}\|u - u_0\|_2^2 \\ \text{s.t.} \quad \dot{y}(f_{1,i}) \quad &= \quad F(y(f_{1,i}), u), \forall i = 1, ..., N. \end{aligned}$$

Here, m and $m_{Climber}$ are vectors of dimension N, where the ith component is the value of the overflow for $f_{1,i}$. $y(f_{1,i})$ is the set of temperatures and salinities for the ith value of f_1, respectively. The dependence of J on the state vector y appears due to the modeling of the overturning $m = m(y)$ in *(Box)*.

The straightforward method to optimize the parameters u is to apply a standard optimization algorithm. For comparison with the Oneshot approach, we use a Quasi-Newton implementation with BFGS update of the Hessian. See chapter 6 for results.

In this case, for each $f_{1,i}$, $i = 1, ..., N$, and each parameter value u during the optimization process the box model has to be run into a steady state, i.e. in our testings $\|y_{k+1} - y_k\| < \varepsilon = 10^{-6}$ for iteration index k. That takes about 400 (for small f_1) up to $11,000$ (for larger f_1) Explicit Euler time steps. A typical Explicit Euler run is illustrated in figure 3.

The required derivatives are AD-generated with the reverse mode of TAF [3]. The reverse mode is chosen to satisfy a bounded computational cost deterioration. That means the cost of an optimization shall be independent of the number of parameters to be optimized. We remark that with only 6 parameters in our optimization problem, the reverse mode is only slightly cheaper than the forward mode, and therefor is not mandatory.

Compared to more complex climate models, here the Euler time step evaluation is not too expensive. However, during the optimization process a large number of Explicit Euler time steps will accumulate and for derivative calculation a huge amount of recomputations, storing or both is necessary. To avoid this, we apply the so-called One-Shot method which is depicted in the following section.

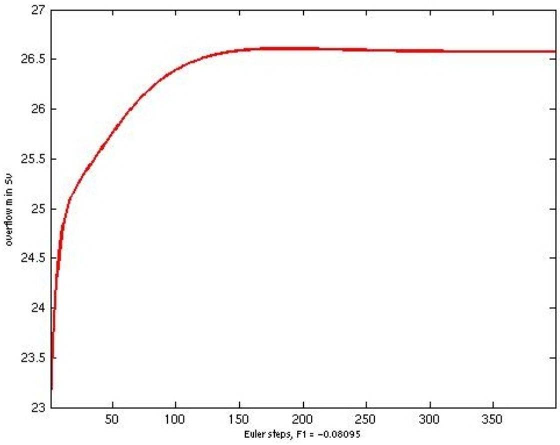

Figure 3: A typical Explicit Euler run

4 The Oneshot Optimization Strategy

In this section we shortly describe the Oneshot method introduced by Griewank and Hamdi [6]. To avoid confusion with the freshwater flux f_1 we denote the cost by J instead of f as in [6]. We consider the optimization problem in general

$$(P) \quad \min_{y,u} J(y,u) \quad s.t. \quad c(y,u) = 0,$$

where $J : Y \times U \to \mathbb{R}$ is the objective function, $u \in U$ the design or parameter vector to be optimized, given in a finite dimensional Hilbert space with $dim(U) = m$. Moreover, $y \in Y$ is the state where Y is a Hilbert space and $c : Y \times U \to Y$ is the state equation. We assume J to be $C^{2,1}$.

The idea is to reformulate the condition $c(y,u) = 0$ into a fixed point equation $y = G(y,u)$ where the iteration function $G : Y \times U \to Y$ is assumed to be $C^{2,1}$ with the contraction factor $\rho < 1$, i.e.

$$\|G(y_1,u) - G(y_2,u)\| \leq \rho \|y_1 - y_2\|, \qquad \forall y_1, y_2 \in Y.$$

In practice, the function G is simply the numerical method originally applied to solve the system $c(y,u) = 0$.

Here and from now on, the subscripts denote the partial derivatives w.r.t y and u, respectively. In the finite dimensional case, the associated Lagrangian to problem (P) with the adjoint $\bar{y} \in Y$ is

$$L(y, \bar{y}, u) = J(y,u) + \bar{y}^\top (G(y,u) - y).$$

$(y_*, \bar{y}_*, u_*)$ fulfills the first order necessary optimality condition, if it holds

$$\begin{array}{rcl}
0 = \frac{\partial L}{\partial y} & = & J_y(y^*, u^*) + \bar{y}^{*\top} G_y(y^*, u^*) - \bar{y}^{*\top}, \\
0 = \frac{\partial L}{\partial \bar{y}} & = & G(y^*, u^*) - y^*, \\
0 = \frac{\partial L}{\partial u} & = & J_u(y^*, u^*) + \bar{y}^{*\top} G_u(y^*, u^*).
\end{array}$$

Herewith, the following iteration is reasonable:

$$\begin{array}{rcll}
y_{k+1} & = & G(y_k, u_k) & \text{to reach primal feasibility,} \\
\bar{y}_{k+1}^T & = & \bar{y}_k^\top G_y(y_k, u_k) + J_y(y_k, u_k) & \text{for dual feasibility, and} \\
u_{k+1} & = & u_k - B_k^{-1} \left(J_u(y_k, u_k) + \bar{y}_k^\top G_u(y_k, u_k) \right) & \text{to obtain optimality.}
\end{array}$$

The update in u resembles a Quasi-Newton step. However, the matrix B must not only provide descent of the Lagrangian, but also convergence of the combination of the three updates. Furthermore, the goal is to achieve bounded retardation which means that convergence of the coupled system shall not be much slower than the convergence of the fixed point iteration G characterized by the contraction factor ρ.

The authors of [6] suggest to choose the preconditioner B as the approximation of the Hessian of the doubly augmented Lagrangian L^a multiplied by a factor σ, where

$$L^a(y,\bar{y},u) = \frac{\alpha_L}{2}\|G(y,u)-y\|^2 + \frac{\beta_L}{2}\|N_y(y,\bar{y},u)^\top - \bar{y}\|^2 + N(y,\bar{y},u) - \bar{y}^\top y,$$

where $\alpha_L > 0$ and $\beta_L > 0$ are weighting coefficients, and N is the shifted Lagrangian $N(y,\bar{y},u) = J(y,u) + \bar{y}^\top G(y,u)$, i.e. $L(y,\bar{y},u) = N(y,\bar{y},u) - \bar{y}^\top y$. The Hessian of L^a is approximately

$$\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu} \approx \nabla^2_{uu} L^a.$$

Hamdi and Griewank prove in [6] that stationary points of problem (P) are also stationary points of L^a. Furthermore, for the latter descent is provided if

$$B = \frac{1}{\sigma}(\alpha_L G_u^\top G_u + \beta_L N_{yu}^\top N_{yu} + N_{uu}) \tag{1}$$

where

$$\sigma := 1 - \rho - \frac{(1 + \frac{\|N_{yy}\|}{2}\beta_L)^2}{\alpha_L \beta_L (1-\rho)}.$$

Since the calculation of second derivatives is very expensive concerning calculation costs and time, in practice, a Low-Rank update, e.g. BFGS update, can be used. It is proposed to choose α_L and β_L as

$$(\alpha_L, \beta_L) = \min_{\alpha,\beta} \frac{\alpha\|G_u\|^2 + \beta\|N_{yu}\|^2}{\sigma}.$$

Under the assumption that $\sqrt{\alpha_L\beta_L}(1-\rho) > 1 + \frac{\beta_L}{2}\|N_{yy}\|$ holds and $\|N_{yy}\| \neq 0$ we obtain

$$\beta_L = \frac{3}{\sqrt{\|N_{yy}\|^2 + 3\frac{\|N_{yu}\|^2}{\|G_u\|^2}(1-\rho)^2} + \frac{\|N_{yy}\|}{2}} \quad \text{and} \quad \alpha_L = \frac{\|N_{yu}\|^2 \beta_L (1 + \frac{\|N_{yy}\|}{2}\beta_L)}{\|G_u\|^2 (1 - \frac{\|N_{yy}\|}{2}\beta_L)}.$$

Griewank and Hamdi make three major assumptions which are the basis of the convergence of the Oneshot method:

1. J is a $C^{2,1}$ function,
2. $\partial c / \partial y$ is always invertible such that for given u there exists only one y with $c(y,u) = 0$ and of course
3. the function $G \in C^{2,1}$ is a fixed point iteration with the contraction factor $\rho < 1$.

5 Oneshot Method for the Advanced Version of the Box Model

Now, we present the realization of the Oneshot method for the advanced box model *(Box)*.

In our investigations, we test the Oneshot method staying very close to the theoretical approach described by Griewank and Hamdi in [6]. That means, we calculate the preconditioner B defined in (1) in every iteration including all first and second order derivatives. It is also possible to use a BFGS update with respect to the doubly augmented Lagrangian L^a to save computational time and costs. As already pointed out in section 3 the minimization problem to be solved is

$$\begin{aligned} \min_{y,u} J(y,u) \quad &:= \quad \frac{1}{2}\|\delta m - m_{Climber}\|_2^2 + \frac{\alpha_w}{2}\|u - u_0\|_2^2 \\ \text{s.t.} \quad \dot{y}(f_{1,i}) \quad &= \quad F(y(f_{1,i}), u), \forall i = 1, \ldots N. \end{aligned}$$

From now on we consider y as the “long” vector $y^\top = (y(f_{1,1})^\top, ..., y(f_{1,N})^\top) \in \mathbb{R}^{8N}$, such that the fixed point function G replacing the constraint by $y = G(y, u)$ is from $\mathbb{R}^{8N}$ to $\mathbb{R}^{8N}$. The Jacobian G_y then is a diagonal block matrix with entries $G_{y(f_{1,i})}$, $i = 1, ..., N$. Analogously, the cost function is of dimension $J : \mathbb{R}^{8N} \times \mathbb{R}^6 \to \mathbb{R}$.

5.1 Verification of Oneshot Strategy Convergence for the Box Model

In chapter 4, we state that there are three major assumptions for the convergence of the Oneshot method. We check their validity in this section.

In our case J as a least squares functional satisfies the regularity assumptions $J \in C^{2,1}$. For the assumed regularity of G it is crucial to use the advanced or smoothed version of the box model *(Box)* which eliminates the kink at states where $m = 0$. Then, we also have that the Explicit Euler time stepping $G(y, u) = y + F(y, u)$ is a $C^{2,1}$ function.

The sufficient and necessary conditions for the function G, to be a contraction are

$$\begin{aligned} \|G_y(y,u)\| &\leq \rho < 1, \quad \forall y \in Y \text{ and} \\ \|G(y_1,u) - G(y_2,u)\| &\leq \rho\|y_1 - y_2\|, \quad \forall y_1, y_2 \in Y, \quad \rho < 1 \end{aligned}$$

respectively.

It was not possible to show that these two conditions hold for all $y \in Y$. In fact, they are not valid for *all* iterates y_k, see figure 5.

However, the Explicit Euler time stepping works for the advanced version of the box model in all our testings. Figure 4 shows Explicit Euler runs for different starting values y_0 and different model parameters u. Even for bad starting values, the Explicit Euler method converges to the same stationary value as with good starting values. Also for bad model parameters the procedure converges.

Considering the ratio

$$\frac{\|G(y_k,u) - G(y_{k-1},u)\|}{\|y_k - y_{k-1}\|} = \frac{\|y_{k+1} - y_k\|}{\|y_k - y_{k-1}\|}$$

in our testings, it is close to 1 for all freshwaterfluxes f_1 and in average less than 1, cp. figure 4. The largest amount of values greater than 1 appears for $f_1 = -0.0259$, which is about 16% of overall 411 iterations. For $f_1 = 0.1341$, the smallest percentage of ratios greater than 1,namely 0.2% of 10,880 iterations, occurs as shown in figure 5. We notice that even thought the contraction property cannot be proven and the ratio even exceeds 1 for several steps in a row in our example, we still find acceptable solutions in all our testings.

As the Explicit Euler time stepping shows slow convergence, one cannot expect the Oneshot method to converge fast in this special example.

The assumption $\partial c/\partial y$ being always invertible is equivalent to G converging to a *unique* limit $y^* = y^*(u)$ for fixed u. As above this assumption could not be proven and even numerical tests show that there are cases, where the $\partial c/\partial y$ is singular. Nevertheless, here too, as shown in figure 4 in practice G converges for fixed u but different starting values y_0 to the same stationary state $y^* = y^*(u)$.

We remark that it is absolutely not unusual in climate modeling that theoretical analysis cannot be provided because of the complexness of real world models. In case of the optimization strategy, its quality and usefulness is even more convincing if good results are achieved even though convergence assumptions are not fulfilled.

5.2 Calculation of Derivatives Using Automatic Differentiation

For the update of the adjoint $\bar{y}$ and model parameters u which requires the determination of the preconditioner B we calculate the necessary first and second order derivatives with the automatic differentiation tool *TAF* which uses the source code transformation approach to generate FORTRAN subroutines to calculate function values and derivative information in one call, see [3].

In contrast to derivative calculation in the BFGS method, where in the reverse mode it is necessary to store all Euler steps until a steady state is reached, here the required derivatives only depend on

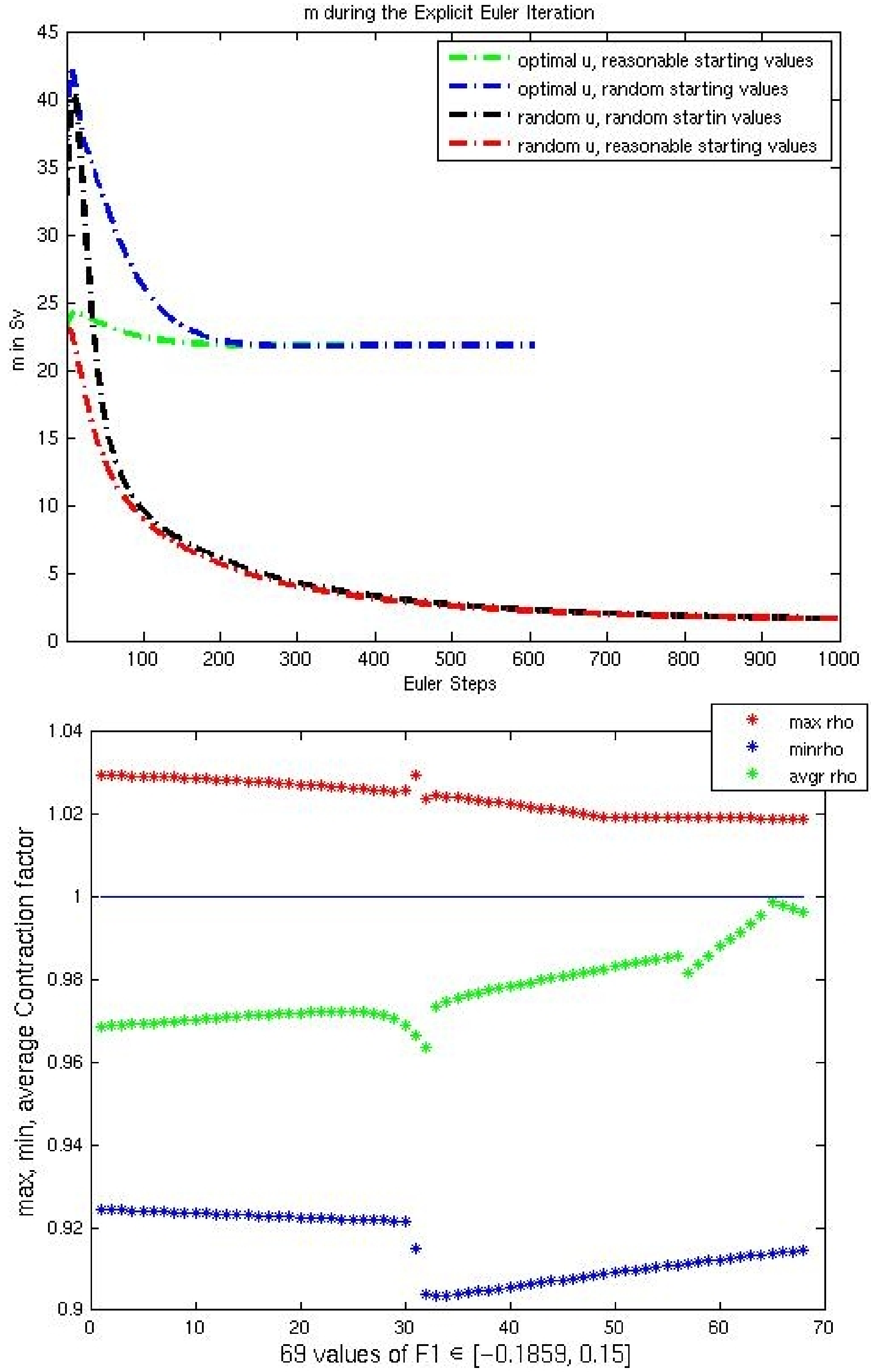

Figure 4: Convergence behaviour of the Explicit Euler time stepping

the current values.
The forward mode of TAF is used for G_u, the reverse mode for $\bar{y}^\top G_y$ and for the second order derivatives $\bar{y}^\top G_{yu}$ and $\bar{y}^\top G_{yy}$ first the reverse and then the forward mode is applied. Derivatives J_u, J_y, J_{yu} and J_{yy} can easily be provided by hand, as done in the appendix.

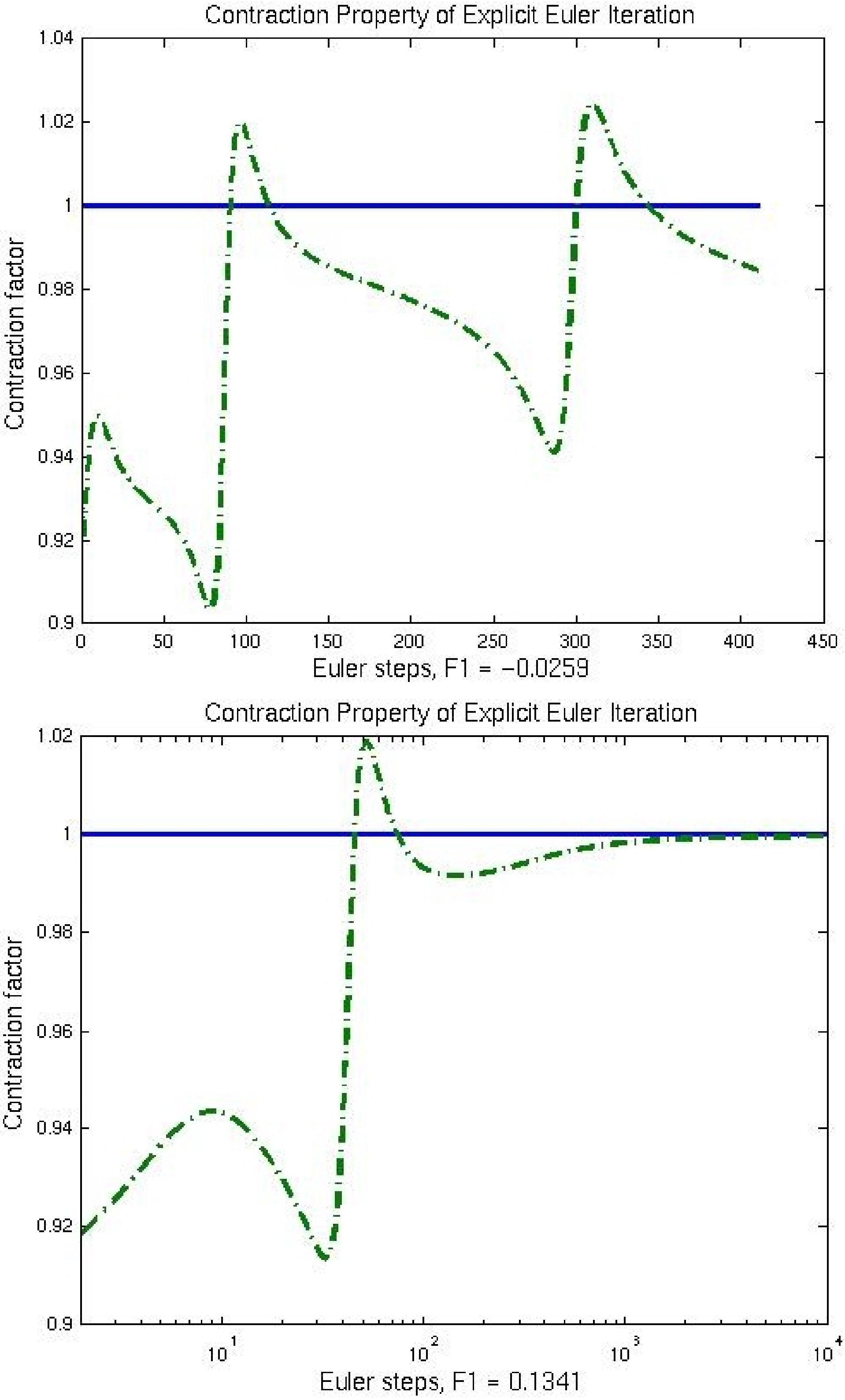

Figure 5: Ratio $\|y_{k+1} - y_k\| / \|y_k - y_{k-1}\|$ for different f_1

6 Numerical Results

In this section, we present numerical results comparing the Oneshot strategy with the popular Quasi-Newton-BFGS method. Before that, we describe some details of the implementation.

6.1 Details of the Implementation

In our test cases we have $N = 68$ different values for freshwater fluxes f_1 in the interval $[-0.18595, 0.15]$ with a spacing of 0.005 between the $f_{1,i}$, $i = 1, ...N$. The overturning $m_{Climber}$ in the Climber model then is between $4.965 * 10^{-8}$ and 29.301 as shown in figure 6. One can see that there are some kinks in the data points, for example between $f_1 \in [0.05, 0.07]$ and $f_1 \in [0.14, 0.15]$. Since the overturning m computed with the boxmodel continuously depend on f_1, an exact fit to those data points will most likely not be achievable.

As proposed in [13], we choose the starting vector $y_0^\top = (y_0(f_{1,1})^\top, ..., y_0(f_{1,N})^\top)$ as the corresponding equilibrium temperatures and salinities derived from the present day overturning $m_{eq} = 22.6$ Sverdrup for all $f_{1,i}$.

We take $u_0 = (6.6, 2.7, 11.7, 23, 25, 500)^\top$ as starting values of the model parameters to be optimized where components 1 to 5 are values close to optimal values of the original box model in [13] and $u(6) = a$ is chosen from tests with the advanced box model which possess good performance.

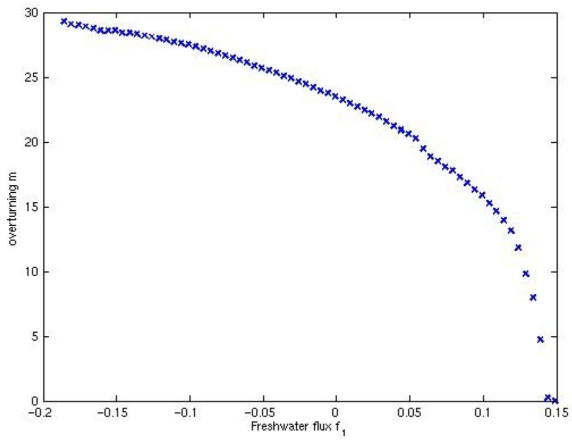

Figure 6: Data from the Climber model

As stopping criterion, we set that the change in the state $y \in \mathbb{R}^{8N}$ is less than $\varepsilon = 10^{-6}$ or the gradient of J is almost 0, i.e. $\|\nabla J\| < \varepsilon$.

For the update of the weighting coefficients α_L, β_L and σ, the contraction factor ρ of the Explicit Euler time stepping is necessary, which we set to a value close to 1, namely $\rho = 0.9$, due to the inability to calculate an exact value (compare subsection 5.1).

Now, we describe, how $\|N_{yy}\|$ is determined for the evaluation of σ, α_L and β_L. We choose the spectral norm $\|\cdot\|_2$ for all matrix norms in our investigations.

On the one hand, we determine $\|G_u\|_2$ and $\|N_{yu}\|_2$ with the help of the Sun Performance Library for Fortran [1] computing the Eigenvalues of $G_u^\top G_u \in \mathbb{R}^{6\times 6}$ and $N_{yu}^\top N_{yu} \in \mathbb{R}^{6\times 6}$.

On the other hand, we apply the *power iteration algorithm*, see [11], to determine the dominant Eigenvalue of the matrix $N_{yy}^\top N_{yy}$. Starting with a random vector b_0, we define

$$\mu_k = \frac{b_k^\top N_{yy}^\top N_{yy} b_k}{b_k^\top b_k} \text{ and } b_{k+1} = \frac{N_{yy}^\top N_{yy} b_k}{\|N_{yy}^\top N_{yy} b_k\|}.$$

If an eigenvalue strictly greater than the other eigenvalues exists and b_0 has a nonzero component in the direction of an eigenvector associated with the dominant eigenvalue, a subsequence of (b_k) converges to that eigenvector associated to μ. Then, it is $\|N_{yy}\|_2 = \sqrt{\mu}$.

In our case, it is $N(y, u) = J(y, u) + \bar{y}^\top G(y, u)$ and therefore

$$N_{yy}(y, u) = J_{yy}(y, u) + \bar{y}^\top G_{yy}(y, u).$$

For the calculation of $N_{yy} b_k$ we determine $\bar{y}^\top G_{yy} b_k$ with a TAF generated subroutine and $J_{yy} b_k$ by hand. Afterwards, a second call of the TAF generated subroutine for the evaluation of $\bar{y}^\top G_{yy}(N_{yy} b_k)$ is necessary to determine $N_{yy}^\top N_{yy} b_k = (J_{yy} + \bar{y}^\top G_{yy})^\top N_{yy} b_k$. Again, $J_{yy}^\top N_{yy} b_k$ can be analyzed by hand.

However, we take a look at the effect of computing $\|N_{yy}\|_2$ in every iteration in our case of the 4-box model: In our testings, the dominant part of $N_{yy}^\top N_{yy}$ is the constant matrix $J_{yy}^\top J_{yy}$ which in our implementation contains entries of range 10^5 (compare appendix), whereas $\bar{y}^\top G_{yy}$ which is a factor in all other summands is of range 10^{-1}. For this reason, $\|N_{yy}\|_2$ does not change significantly from iteration to iteration and we test how the Oneshot strategy reacts if the power iteration is applied only every 1000 and every 10000 steps. Therefore, an inaccuracy in the computation of σ, α_L and β_l is the result. In our case, as one can see in table 1, this simplification does not significantly influence the optimization, but saves calculation time.

	#iterations	time in minutes	$J(y^*, u^*)$	fit
every 10000 iterations	1,037,804	6.174	14.879	14.053
every 1000 iterations	1,010,011	6.148	14.879	14.053
every iteration	1,015,563	10.394	14.879	14.053

Table 1: Application of the power iteration not in every iteration and its effect on the Oneshot optimization for $\alpha_w = 0.1$

In our calculations the weighting coefficient β_L becomes very small in the range of 10^{-5} whereas α_L then is large, i.e. in the range of 10^5. Since $N_{yu}^\top N_{yu}$ contains quite large values and $G_u^\top G_u$ only small ones, we can assume that the weighting coefficients are well chosen for the update of B.
Therewith, we can now investigate the behaviour of the Oneshot optimization strategy applied to the 4-box-model.

6.2 Comparison Between the BFGS and Oneshot Strategies for Different α_w

In this section, we describe our observations concerning the optimization behaviour with respect to the choice of the factor α_w in front of the penalty term $\|u - u_0\|$. It is of crucial importance for the convergence and the result of the optimization.
First of all, we find that α_w must not be 0, because otherwise the Oneshot method does not converge. The BFGS method finds optimal values u^* with $\frac{\partial J}{\partial u}(u^*) = 0$, but those are too far away from reality. Not surprisingly, one detects that the smaller α_w the better the fit of the data becomes. In figure 7 we can see that especially for large f_1 the fit becomes better and also for values in front of the kink at $f_1 \approx 0.06$.

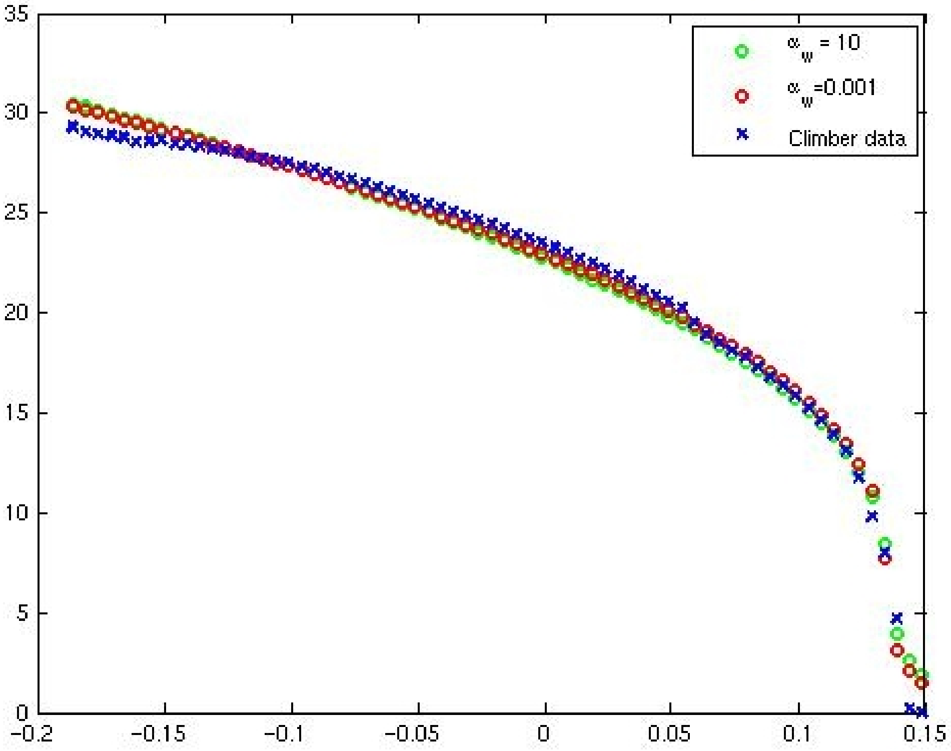

Figure 7: Results computed by the Oneshot method for different α_w

Figure 8 illustrates a comparison between the fit provided by the Oneshot method and BFGS. The results differ only slightly. Whereas for $\alpha_w = 0.001$ the BFGS method fits data better for small f_1, the Oneshot strategy reaches better results for large f_1. In the case $\alpha_w = 10$ there is only one data point where the fit differs between the two optimization strategies, namely the Oneshot method achieving a better value.

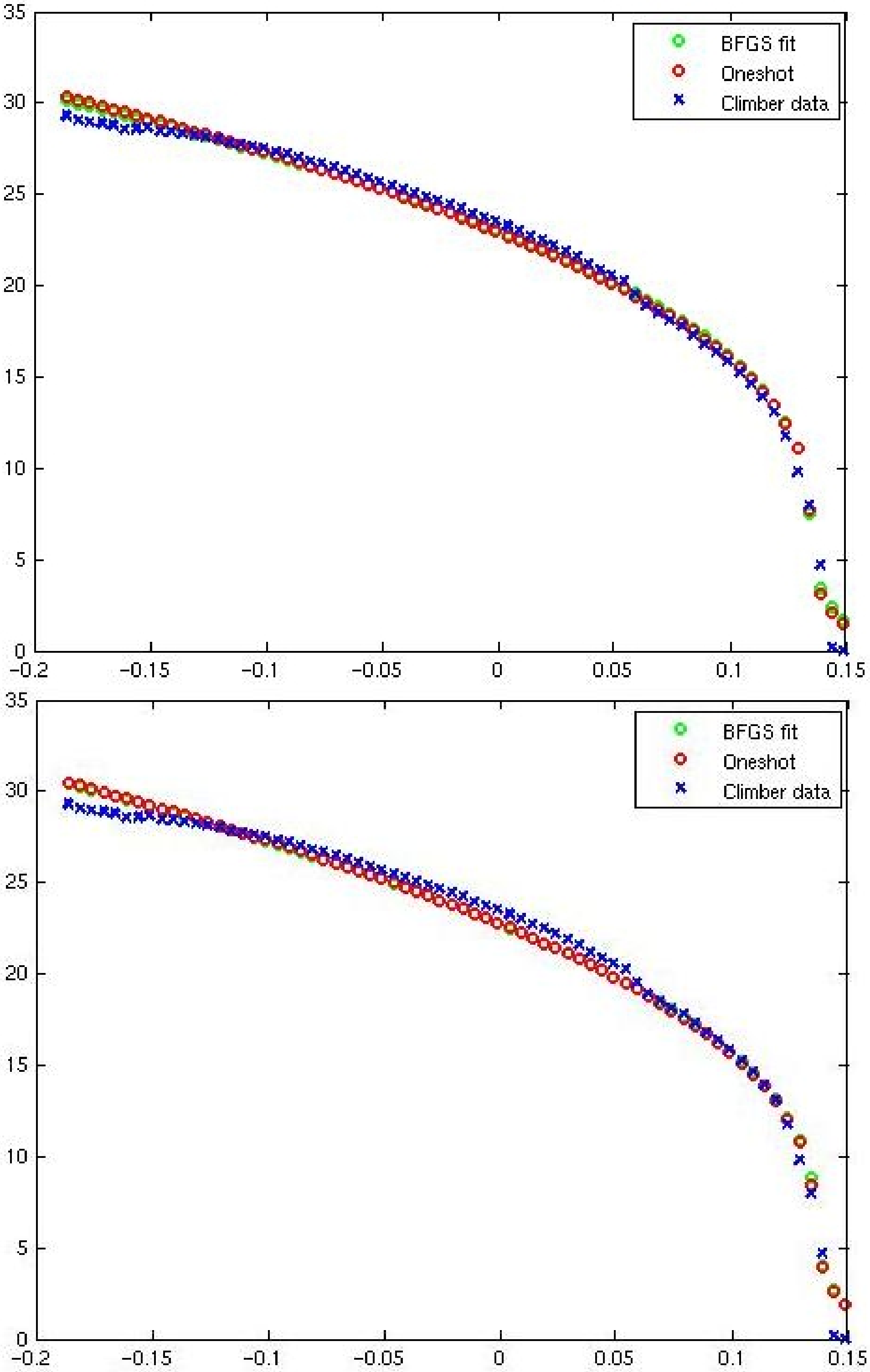

Figure 8: Results of the optimization comparing the Oneshot strategy with the BFGS method. At the top for $\alpha_w = 0.001$, at the bottom for $\alpha_w = 10$.

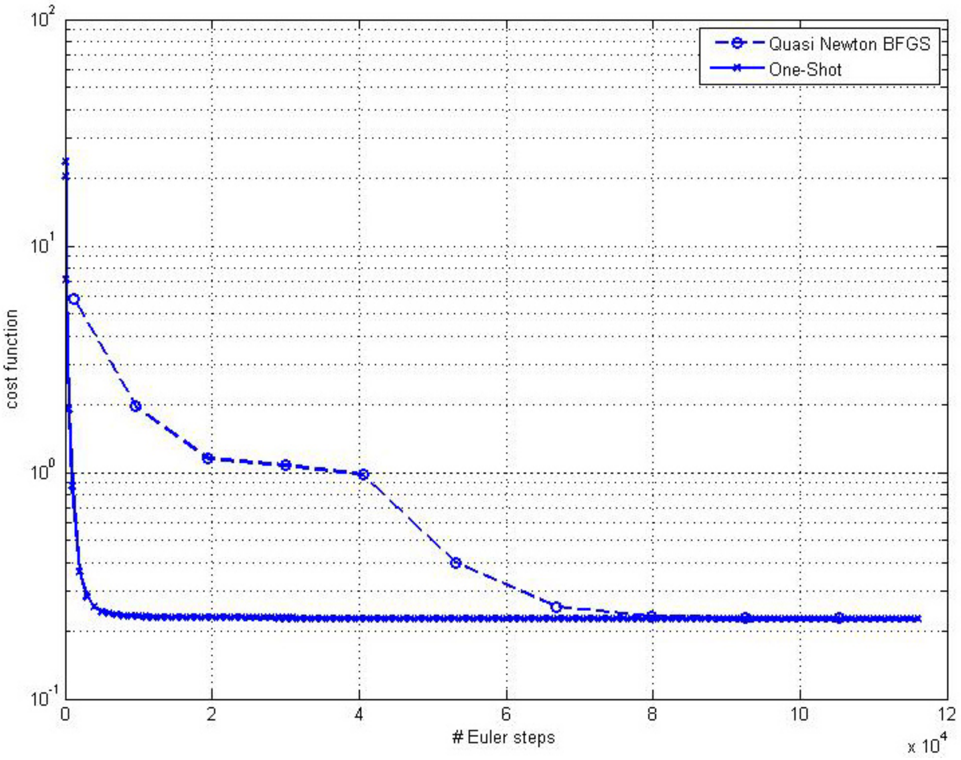

Figure 9: The cost function $J(y, u)$ w.r.t. the number of Euler steps.

Table 2 lists our observations concerning the function value (including the penalty term) at optimal values found by the two optimization strategies, the fit averaged over the number of data points, the number of iterations, and the overall number of Euler steps. We obtain better results for $\alpha_w = 10$ and $\alpha_w = 0.1$ with the Oneshot strategy, whereas BFGS finds a better result for $\alpha_w = 0.001$. That confirms again, that the weighting coefficient in front of the penalty term must not be too small otherwise the Oneshot method has difficulties in finding acceptable model parameters.

	$\alpha_w = 10$		$\alpha_w = 0.1$		$\alpha_w = 0.001$	
	Oneshot	BFGS	Oneshot	BFGS	Oneshot	BFGS
$J(y^*, u^*)$	25.854	26.221	14.879	15.926	12.748	11.411
data fit	0.269	0.277	0.206	0.213	0.183	0.166
# iterations	1,269,019	20	1,010,011	28	10,678,000	65
# Euler steps	1,269,019	1,285,203	1,010,011	1,808,823	10,678,000	4,236,481

Table 2: Results of the optimization

Concerning computational time, the BFGS method is a little bit faster and only in the case where $\alpha_w = 0.001$ even significantly with 8 minutes compared to 45 minutes. We remark, that the Oneshot stopping criterion, namely that the change in the "long" vector $y \in \mathbb{R}^{544}$ is less than $\varepsilon = 10^{-6}$, is stricter than the Euler stopping criterion within the BFGS method where for each f_i, $i = 1, ..., 68$, the change in the "short" vectors $y_i \in \mathbb{R}^8$ is less than $\varepsilon = 10^{-6}$. That means, the BFGS method saves Euler steps and therewith computational time, but all in all it still needs more Euler steps than the Oneshot method. Furthermore, the Oneshot strategy is much faster close to the solution than BFGS, as illustrated in figure 9, and one could save iterations mitigating the stopping criterion.

6.3 Bounded Retardation

The goal was to achieve so-called bounded retardation for the speed of convergence. Since the Explicit Euler time stepping does not show quick convergence and ratios $\|y_{k+1} - y_k\|/\|y_k - y_{k-1}\|$ even exceed 1 for several steps, one cannot expect the Oneshot method to converge very fast in this special example. In our testing, the Oneshot method shows a similar behaviour as the Explicit Euler time stepping.

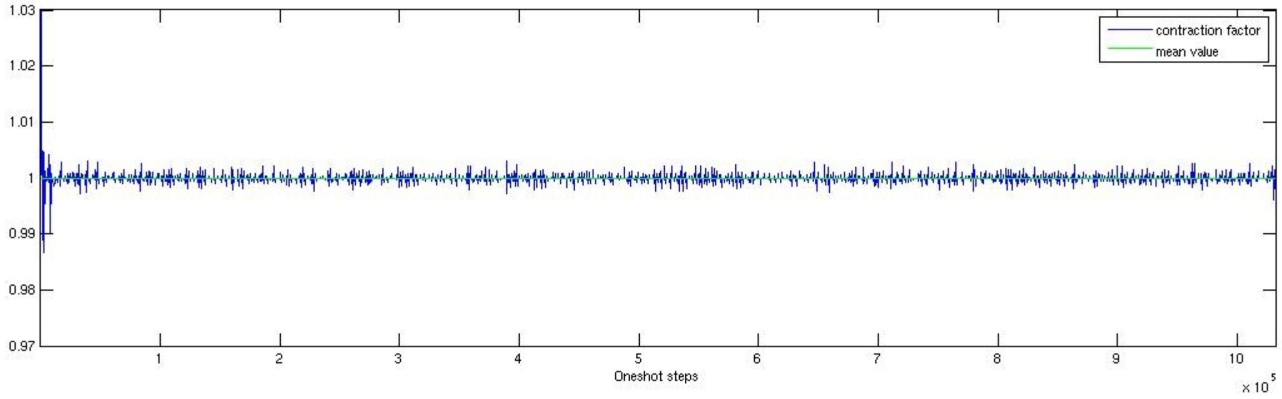

Figure 10: Ratio $\|y_{k+1} - y_k\| / \|y_k - y_{k-1}\|$ during the Oneshot optimization for $\alpha_w \approx 0.1$

The ratio $\|y_{k+1} - y_k\| / \|y_k - y_{k-1}\|$ is close to 1 as can be seen in figure 10.

In this figure, it seems that the ratio is larger than or equal to 1 in about 50% of all iterations, but that is the case in only 4.1%. However, the ratio is smaller than 0.9999 in only 2.2% respectively. In most cases it is between 0.9999 and 1, 0.9999884 in average, which cannot be visualized in the plot.

7 Summary and Outlook

In this paper, we numerically solved a parameter identification problem of a box model of the north Atlantic Thermohaline Circulation. The applied Oneshot methodology proved to be a favourable tool even though the assumptions for convergence cannot be verified analytically, which is not unusual in climate modeling. In comparison to the BFGS method, the obtained optimum was slightly better and the total number of needed time steps was reduced in the case where the penalty term $\frac{\alpha_w}{2}\|u - u_0\|$ was not too small.

Furthermore, we achieved a speed of convergence of the Oneshot method which is not much slowed down compared to the speed of the model spin up only.

Here in this optimization problem, parameters entered in a non-linear fashion resulting in non-separable adjoints. Derivatives were computed with the automatic differentiation tool TAF, which fast and effectively computed function values and derivative information in one call.

Improvements could be made by applying a linesearch procedure which then might also improve the behaviour in tests where the penalty term is very small or good starting values are unknown.

We conclude that the Oneshot approach was successful and represents a useful tool which we can apply to more more complex climate models in our future test.

Acknowledgements

The authors would like to thank Adel Hamdi, Andreas Griewank, Stefan Rahmstorf, and Kirsten Zickfeld for support.

A Derivatives of $J(y,u)$

The cost function $J:\mathbb{R}^{8N\times 6}\to\mathbb{R}$ is defined by

$$J(y,u) \quad := \quad \frac{1}{2}\|\delta m - m_{Climber}\|_2^2 + \frac{\alpha_w}{2}\|u-u_0\|_2^2$$

where $m=(m(y(f_{1,i})))_{i=1,\ldots,N}\in\mathbb{R}^N$, $u=(T_1^*,T_2^*,T_3^*,\Gamma,k,a)$. For simplicity, we abbreviate $m_i := m(y(f_{1,i}))$ and $\Delta_i := \delta m_i - m_{Climber,i}$

$$\begin{aligned}
\frac{\partial J}{\partial y} &= \left(\Delta_i \frac{\delta\partial m_i}{\partial y(f_{1,i})}\right)_{i=1,\ldots,N} \\
&= \left(\Delta_i\delta\begin{pmatrix}\alpha k\\ -\alpha k\\ 0\\ 0\\ -\beta k\\ \beta k\\ 0\\ 0\end{pmatrix}\right)_{i=1,\ldots,N} \qquad \in\mathbb{R}^{8N}. \\
\frac{\partial J}{\partial u} &= \begin{pmatrix}0\\ 0\\ 0\\ 0\\ \sum_{i=1}^N \Delta_i\frac{\delta m_i}{k}\\ 0\end{pmatrix} + \alpha_w\begin{pmatrix}u(1)-u_0(1)\\ u(2)-u_0(2)\\ u(3)-u_0(3)\\ u(4)-u_0(4)\\ u(5)-u_0(5)\\ u(6)-u_0(6)\end{pmatrix} \in\mathbb{R}^6.
\end{aligned}$$

$\frac{\partial^2 J}{\partial y\partial u}\in\mathbb{R}^{8N\times 6}$ consists of N matrices $A_i\in\mathbb{R}^{8\times 6}$ with

$$A_i \quad = \quad \begin{pmatrix}
0 & 0 & 0 & 0 & \delta\alpha\Delta_i+\delta^2\alpha m_i & 0\\
0 & 0 & 0 & 0 & -\delta\alpha\Delta_i-\delta^2\alpha m_i & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -\delta\beta\Delta_i-\delta^2\beta m_i & 0\\
0 & 0 & 0 & 0 & \delta\beta\Delta_i+\delta^2\beta m_i & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

$\frac{\partial^2 J}{\partial^2 y}\in\mathbb{R}^{8N\times 8N}$ is a diagonal block matrix with entries $B_i\in\mathbb{R}^{8\times 8}$ where

$$B_i \quad = \quad \delta^2\begin{pmatrix}
k^2\alpha^2 & -k^2\alpha^2 & 0 & 0 & -k^2\alpha\beta & k^2\alpha\beta & 0 & 0\\
-k^2\alpha^2 & k^2\alpha^2 & 0 & 0 & k^2\alpha\beta & -k^2\alpha\beta & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-k^2\alpha\beta & k^2\alpha\beta & 0 & 0 & k^2\beta^2 & -k^2\beta^2 & 0 & 0\\
k^2\alpha\beta & -k^2\alpha\beta & 0 & 0 & -k^2\beta^2 & k^2\beta^2 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

References

[1] Sun performance library user's guide, 2000. available at `http://docs.sun.com/source/806-3566/plug_title.html`.

[2] Christianson, B. Reverse Accumulation and Implicit Functions. *Optimization Methods and Software 9*, 4 (1998), 307–322.

[3] Giering, R., Kaminski, T., and Slawig, T. Generating efficient derivative code with TAF: Adjoint and tangent linear Euler flow around an airfoil. *Future Generation Computer Systems 21*, 8 (2005), 1345–1355.

[4] Griewank, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* No. 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.

[5] Griewank, A., and Hamdi, A. Properties of an augmented lagrangian for design optimization. *DFG - SPP1253 Preprint-Nr.: SPP1253-11-01* (2008). available at `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`.

[6] Griewank, A., and Hamdi, A. Reduced quasi-newton method for simultaneous design and optimization. *DFG - SPP1253 Preprint-Nr.: SPP1253-11-02* (2008). available at `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`.

[7] Kaminski, T., Giering, R., and beck, M. V. Efficient sensitivities for the spin-up phase. In *Automatic Differentiation: Applications, Theory, and Implementations* (2005), H. M. Bcker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, Eds., Springer, New York, pp. 283–291. volume 50 of Lecture Notes in Computational Science and Engineering.

[8] Kratzenstein, C., and Slawig, T. One-shot parameter optimization in a box-model of the north atlantic thermohaline circulation. *DFG - SPP1253 Preprint-Nr.: SPP1253-11-03* (2008). available at `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`.

[9] Özkaya, E., and Gauger, N. Single-step one-shot aerodynamic shape optimization. *DFG - SPP1253 Preprint-Nr.: SPP1253-10-04* (2008). available at `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`.

[10] Rahmstorf, S., Brovkin, V., Claussen, M., and Kubatzki, C. Climber-2: A climate system model of intermediate complexity. part ii: Model sensitivity. *Clim. Dyn. 17* (2001), 735–751.

[11] Stoer, J., and Bulirsch, R. *Introduction to Numerical Analysis.* Springer-Verlag, New York and Berlin, 1980.

[12] Titz, S., Kuhlbrodt, T., Rahmstorf, S., and Feudel, U. On freshwater-dependent bifurcations in box models of the interhemispheric thermohaline circulation. *Tellus A 54* (2002), 89 – 98.

[13] Zickfeld, K., Slawig, T., and Rahmstorf, S. A low-order model for the response of the atlantic thermohaline circulation to climate change. *Ocean Dynamics 54* (2004), 8–26.

B Model Spin-up function and its derivatives for the N-DOP model with fixed Transport Matrices

In this section we provide a very short overview of the N-DOP model, a model of the global phosphorus cycle, using notations from Prieß et. al. [32] who refers to Kriest et. al. [23] for details. A mathematical analysis of the model concerning the existence of solutions is provided by Roschat et al. in [35] and [36].
As in the main part of this work, $u \in U$ denotes the model parameters to be optimized and $y \in Y$ the state variables for which we intend to find a steady annual cycle.
The N-DOP model investigates two tracers phosphorus y_N and dissolved organic phosphorus y_{DOP}. We define

$$\begin{aligned} u &= (\lambda, \alpha, \sigma, K_N, K_I, K_{H2O}, b) \in \mathbb{R}^7; && \text{(B.1)} \\ y &= (y_N, y_{DOP}) \in Y \subset \mathbb{R}^n \text{ where } n = dim(y_N) + dim(y_{DOP}). && \text{(B.2)} \end{aligned}$$

A general biogeochemical ocean model describes the tracer concentrations $y = y(x, t)$ on a space-time cylinder, $(x, t) \in \Omega \times [0, T]$ with $\Omega \in \mathbb{R}^3$ being the spacial domain, $\Gamma = \partial\Omega$ its boundary, and $[0, T]$, $T > 0$ the time interval. The model equations consist of an advection, diffusion and tracer coupling term for each tracer y_i

$$\frac{\partial y_i}{\partial t} = \underbrace{\nabla \cdot (\kappa \nabla y_i)}_{\text{diffusion}} - \underbrace{\nabla \cdot (v y_i)}_{\text{advection}} + \underbrace{q_i(y, u)}_{\substack{\text{non-linear,} \\ \text{non-separable coupling}}}$$

where $v = v(x, t)$ is the advection velocity vector field mixing coefficient $\kappa = \kappa(x, t)$, temperature and salinity. Usually, homogeneous Neumann boundary conditions on Γ are imposed for the tracers y_i, $i \in \{N, DOP\}$.
The model we work with is an off-line model which simplifies the structure of the simulation of tracers in that the tracers are regarded as passive meaning that they do

not affect the ocean physics. Here, the ocean circulation data enters the tracer transport equation as annually fixed forcing terms varying during the period of one year. It is precomputed and stored in so called Transport Matrices $A_{imp,j}$ and $A_{exp,j}$ for each intermediate time integration step $j = 0, ..., n_t - 1$, where n_t denotes the number of time steps per year. The Transport Matrix Method was introduced by Khatiwala et al. [19].

Taking these fixed linear mappings simplifies the problem in that searching for an annually periodic solution is equivalent to seeking for steady tracers at one (arbitrary) time point of the year.

The resulting time integration scheme or model spin-up function, which we denote as G in the main part, $y_{k+1} = G(y_k, u)$, $k \geq 0$, consists of $n_t = 2,880$ intermediate time steps per model year and is defined as

$$\begin{aligned} y_{k+1} &= y_{k+1,0} := y_{k,n_t} \text{ where} && \text{(B.3)} \\ y_{k,j+1} &= A_{imp,j}(A_{exp,j} y_{k,j} + q_j(y_{k,j}, u)), \text{ for } j = 0, ..., n_t - 1. && \text{(B.4)} \end{aligned}$$

q_j is the coupling function of source-minus-sink type which depends on the parameter and state variables, but it also depends on the intensity of light I which varies in latitude and season and therefore we write $q_j(y, u)$ in the evaluation of the model spin-up function G to imply the seasonal dependency.

In the following description of the model we omit the subscripts j and k for better readability.

For the two tracers we consider two parts of q

$$q(y, u) = (q_N(y, u), q_{DOP}(y, u)) = (q_N(y_N, y_{DOP}, u), q_{DOP}(y_N, y_{DOP}, u)) \quad \text{(B.5)}$$

Different equations for the different parts of the ocean hold. Therefore, the spacial domain $\Omega \in \mathbb{R}^3$ is divided into an euphotic (sun lit) zone Ω_1 and the deeper ocean, aphotic (or non-euphotic) zone Ω_2.

The biological source-minus-sink terms then read

$$q_N(y_N, y_{DOP}, u) = \begin{cases} -f(y_N, I) + \lambda\, y_{DOP} & \text{in } \Omega_1 \\ (1 - \sigma) \frac{\partial}{\partial z} F(y_N, I) + \lambda\, y_{DOP} & \text{in } \Omega_2 \end{cases} \quad \text{(B.6)}$$

$$q_{DOP}(y_N, y_{DOP}, u) = \begin{cases} \sigma\, f(y_N, I) - \lambda\, y_{DOP} & \text{in } \Omega_1 \\ -\lambda\, y_{DOP} & \text{in } \Omega_2 \end{cases} \quad \text{(B.7)}$$

where

$$F(y_N, I) = \left(\frac{z}{z'}\right)^{-b} \int_0^{z'} f(y_N, I)\, dz\,.$$

the biological production is

$$f(y_N, I) = \alpha\, \frac{y_N}{y_N + K_N}\, \frac{I}{I + K_I}\,.$$

and the light I

$$I = I_{SWR}\, \sigma_{PAR}\, (1 - \sigma_{ice})\, \exp(-z\, K_{H2O})\,,$$

where z denotes the vertical coordinate. z' is the depth of Ω_1. I_{SWR}, σ_{PAR} and σ_{ice} are known and vary seasonally.
Inserting $\frac{\partial}{\partial z}\, F(y_N, I) = \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} f(y_N, I)\, dz$ yields

$$q_N(y_N, y_{DOP}, u) \quad = \quad \begin{cases} -f(y_N, I) + \lambda\, y_{DOP} & \text{in } \Omega_1 \\ (1-\sigma)\, \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} f(y_N, I)\, dz + \lambda\, y_{DOP} & \text{in } \Omega_2. \end{cases}$$

We do not go into detail about the meaning of the parameters. We provide an overview over the model equations, the computation of derivatives and we intend to complement the statements on the importance and influence of parameter $u_7 = b$ on the system of equations and on the functioning of the One-shot strategy.

In the following we provide the partial derivatives of q needed for the computation of the derivatives of the model spin-up function G.

$$\frac{\partial}{\partial u} q_N = \begin{pmatrix} y_{DOP} \\ -\frac{y_N}{y_N + K_N}\, \frac{I}{I + K_I} \\ 0 \\ \alpha \frac{y_N}{(y_N + K_N)^2}\, \frac{I}{I + K_I} \\ \alpha \frac{y_N}{y_N + K_N}\, \frac{I}{(I + K_I)^2} \\ 0 \end{pmatrix}, \text{ for } y \in \Omega_1,$$

$$\frac{\partial}{\partial u} q_N = \begin{pmatrix} y_{DOP} \\ (1-\sigma)\ \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} \frac{y_N}{y_N+K_N}\ \frac{I}{I+K_I}\, dz \\ -\ \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} \frac{\alpha y_N}{y_N+K_N}\ \frac{I}{I+K_I}\, dz \\ -(1-\sigma)\ \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} \frac{\alpha y_N}{(y_N+K_N)^2}\ \frac{I}{I+K_I}\, dz \\ -(1-\sigma)\ \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} \frac{\alpha y_N}{y_N+K_N}\ \frac{I}{(I+K_I)^2}\, dz \\ -(1-\sigma)\ \left(\frac{z}{z'}\right)^{-b-1} \left(\frac{-b}{z'}\right) \int_0^{z'} \frac{\alpha y_N}{y_N+K_N}\ \frac{K_I z I}{(I+K_I)^2}\, dz \\ -\frac{(1-\sigma)}{z'}\ \left(\frac{z}{z'}\right)^{-b-1} \left(1 - b\ln\left(\frac{z}{z'}\right)\right) \int_0^{z'} \frac{\alpha y_N}{y_N+K_N}\ \frac{I}{I+K_I}\, dz \end{pmatrix}, \text{ for } y \in \Omega_2,$$

$$\frac{\partial}{\partial u} q_{DOP} = \begin{pmatrix} -y_{DOP} \\ \sigma \frac{y_N}{y_N+K_N}\ \frac{I}{I+K_I} \\ \frac{\alpha y_N}{y_N+K_N}\ \frac{I}{I+K_I} \\ -\sigma \frac{\alpha y_N}{(y_N+K_N)^2}\ \frac{I}{I+K_I} \\ -\sigma \frac{\alpha y_N}{y_N+K_N}\ \frac{I}{(I+K_I)^2} \\ -\sigma \frac{\alpha y_N}{y_N+K_N}\ \frac{I K_I z}{(I+K_I)^2} \\ 0 \end{pmatrix}, \text{ for } y \in \Omega_1 \text{ and}$$

$$\frac{\partial}{\partial u} q_{DOP} = \begin{pmatrix} -y_{DOP} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \text{ for } y \in \Omega_2.$$

The necessary derivatives with respect to y are as follows:

$$\begin{aligned} \frac{\partial q_N}{\partial y} &= \begin{pmatrix} \frac{\partial q_N}{\partial y_N} \\ \frac{\partial q_N}{\partial y_{DOP}} \end{pmatrix} = \begin{pmatrix} -\frac{\partial f(y_N,I)}{\partial y_N} \\ \lambda \end{pmatrix} \text{ for } y \in \Omega_1 \text{ and} \\ \frac{\partial q_N}{\partial y} &= \begin{pmatrix} (1-\sigma)\left(\frac{z}{z'}\right)^{-b-1} \frac{-b}{z'} \int_0^{z'} \frac{\partial f(y_N,I)}{\partial y_N} dz \\ \lambda \end{pmatrix} \text{ for } y \in \Omega_2 \end{aligned}$$

with

$$\frac{\partial f(y_N, I)}{\partial y_N} = \alpha \frac{I}{I+K_I} \frac{K_N}{(y_N+K_N)^2},$$

and

$$\begin{aligned}
\frac{\partial q_{DOP}}{\partial y} &= \begin{pmatrix} \frac{\partial q_{DOP}}{\partial y_N} \\ \frac{\partial q_{DOP}}{\partial y_{DOP}} \end{pmatrix} = \begin{pmatrix} \sigma \frac{\partial f(y_N, I)}{\partial y_N} \\ -\lambda \end{pmatrix} \text{ for } y \in \Omega_1 \text{ and} \\
\frac{\partial q_{DOP}}{\partial y} &= \begin{pmatrix} 0 \\ -\lambda \end{pmatrix} \text{ for } y \in \Omega_2.
\end{aligned}$$

In chapter 4.3.3 of the main part of this work, we mentioned parameter $u_7 = b$ responsible for a completely different working flow of the One-shot strategy in that the inclusion of b into the optimization problem introduces a new structure of derivatives. Without u_7 to be optimized it was $u_6 = K_{H2O}$ having the main influence. Obviously, u_7 has a large impact on the other parameter corrections as it has a major impact on the size of the entries of $\frac{\partial}{\partial u} q_N$ in Ω_2 and even the signs of the entries of $\frac{\partial}{\partial u_7} q_N$ in Ω_2 which may change with varying b. A too big correction in u_7 due to an inexactly chosen preconditioner B inevitably implicates a too large correction in the other components.

For completeness, we list the algorithms to compute the full Jacobian of G carried forward over the length of the period of one year with n_t intermediate time steps. Applying the model evaluation function G defined in (B.3) the computation of $G_y(y_k, u)$ and $G_u(y_k, u)$ in the One-shot iteration step k, where $y_{k,0}$ is regarded independent from u_k, then is of the following structure

$$\begin{aligned}
\frac{\partial}{\partial u} G(y_k, u) &:= \frac{\partial}{\partial u} y_{k,n_t}, \text{ where } \frac{\partial}{\partial u} y_{k,0} = 0 \text{ and} \\
\frac{\partial}{\partial u} y_{k,j+1} &= A_{imp,j}(A_{exp,j} \frac{\partial}{\partial u} y_{k,j} + \frac{\partial}{\partial y} q_j(y_{k,j}, u) \frac{\partial}{\partial u} y_{k,j} + \frac{\partial}{\partial u} q_j(y_{k,j}, u)), \\
&\quad \text{for } j = 0, ..., n_t - 1. \\
\frac{\partial}{\partial y} G(y_k, u) &:= \frac{\partial}{\partial y} y_{k,n_t} \text{ where } \frac{\partial}{\partial y} y_{k,0} = 1 \text{ and} \\
\frac{\partial}{\partial y} y_{k,j+1} &= A_{imp,j}(A_{exp,j} \frac{\partial}{\partial y} y_{k,j} + \frac{\partial}{\partial y} q_j(y_{k,j}, u) \frac{\partial}{\partial y} y_{k,j}), \text{ for } j = 0, ..., n_t - 1.
\end{aligned}$$

Here, we use the subscript j of q_j again implying the seasonal dependency of $q(y, u)$ on light I.

Bibliography

[1] Balay, S., Brown, J., , Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.

[2] Bosse, T., Gauger, N., Griewank, A., Günther, S., Kaland, L., Kratzenstein, C., Lehmann, L., Nemili, A., Özkaya, E., and Slawig, T. Optimal Design with Bounded Retardation for Problems with Non-separable Adjoints. In *Trends in PDE Constrained Optimization*, G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, Eds., vol. 165 of *International Series of Numerical Mathematics*. Birkhäuser Basel, 2014.

[3] Boyer, T., Antonov, J., Baranova, O., Coleman, C., Garcia, H., Grodsky, A., Johnson, D., Locarnini, R., Mishonov, A., O'Brien, T., Paver, C., Reagan, J., Seidov, D., Smolyar, I., and Zweng, M. World ocean database 2013. Tech. rep., NOAA Atlas NESDIS 72, 2013. S. Levitus, Ed.; A. Mishonov, Technical Ed.

[4] Christianson, B. Reverse Accumulation and Implicit Functions. *Optimization Methods and Software 9*, 4 (1998), 307–322.

[5] Ciric, L. B. A generalization of Banach's contraction principle. *Proceedings of the American Mathematical Society 45*, 2 (1974), 267–273.

[6] Garcia, H. E., Locarnini, R. A., Boyer, T. P., and Antonov, J. I. World ocean atlas, vol. 4, nutrients (phosphate, nitrate, silicate). *NOAA Atlas NESDIS 64* (2006).

[7] Gauger, N., Griewank, A., Hamdi, A., Kratzenstein, C., Özkaya, E., and Slawig, T. Automated Extension of Fixed Point PDE Solvers for Optimal Design with Bounded Retardation. In *Constrained Optimization and Optimal*

Control for Partial Differential Equations, G. Leugering, S. Engel, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, and S. Ulbrich, Eds., vol. 160 of *International Series of Numerical Mathematics*. Birkhäuser Basel, 2011.

[8] Giering, R., and Kaminski, T. Applying TAF to generate efficient derivative code of Fortran 77-95 programs. *Proceedings in Applied Mathematics and Mechanics 2*, 1 (2003), 54–57.

[9] Golub, G. H., and Van Loan, C. F. *Matrix computations (3rd ed.).* Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[10] Griewank, A., and Hamdi, A. Properties of an augmented Lagrangian for design optimization. *Optimization Methods and Software 25*, 4 (2010), 645–664.

[11] Griewank, A., and Hamdi, A. Reduced quasi-Newton method for simultaneous design and optimization. *Computational Optimization and Applications Online 49* (2011), 521–548.

[12] Griewank, A., Juedes, D., and Utke, J. Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software 22*, 2 (1996), 131–167.

[13] Griewank, A., and Kressner, D. Time-lag in derivative convergence for fixed point iterations. *ARIMA Numéro spécial CARI'04* (2005), 87–102.

[14] Griewank, A., and Walther, A. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Society for Industrial and Applied Mathematics (SIAM), 2008.

[15] Günther, S., Gauger, N., and Wang, Q. Extension of the one-shot method for optimal control with unsteady pdes. In *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, D. Greiner, B. Galván, J. Périaux, N. Gauger, K. Giannakoglou, and G. Winter, Eds., Computational Methods in Applied Sciences. Springer International Publishing, 2015, pp. 127–142.

[16] Hazra, S. B., and Schulz, V. Simultaneous pseudo-timestepping for pde-model based optimization problems. *BIT Numerical Mathematics 44* (2004), 457–472.

[17] H.Bischof, C., Lang, B., and Vehreschild, A. Automatic differentiation for MATLAB programs. *Proceedings in Applied Mathematics and Mechanics 2*, 1 (2003), 50–53.

[18] Kaminski, T., Giering, R., and Vossbeck, M. Efficient sensitivities for the spin-up phase. In *Automatic Differentiation: Applications, Theory, and Implementations*, H. M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, Eds., Lecture Notes in Computational Science and Engineering. Springer, 2005, pp. 285–293.

[19] Khatiwala, S., Visbeck, M., and Cane, M. Accelerated simulation of passive tracers in ocean circulation models. *Ocean Modelling 9*, 1 (2005), 51–69.

[20] Kratzenstein, C., and Slawig, T. One-shot parameter optimization in a box-model of the north atlantic thermohaline circulation. Tech. Rep. Preprint SPP1253-11-03, DFG priority program 1253, `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`, 2008.

[21] Kratzenstein, C., and Slawig, T. Oneshot parameter identification - simultaneous model spin-up and parameter optimization in a box model of the north atlantic thermohaline circulation. Tech. Rep. Preprint 082, DFG priority program 1253, `http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`, 2009.

[22] Kratzenstein, C., and Slawig, T. Simultaneous model spin-up and parameter identification with the one-shot method in a climate model example. *International Journal of Optimization and Control: Theory and Applications 3*, 2 (2013), 99–110.

[23] Kriest, I., Khatiwala, S., and Oschlies, A. Towards an assessment of simple global marine biogeochemical models of different complexity. *Progress In Oceanography 86*, 3-4 (2010), 337–360.

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

[25] Nocedal, J., and Wright, S. J. *Numerical Optimization*, 2nd ed. Springer, New York, 2006.

[26] Özkaya, E., and Gauger, N. Single-step one-shot aerodynamic shape optimization. *International Series of Numerical Mathematics 158* (2009), 191–204.

[27] Papageorgiou, N., and Kyritsi-Yiallourou, S. *Handbook of Applied Analysis.* Advances in Mechanics and Mathematics. Springer, 2009.

[28] Pernice, M., and Walker, H. F. Nitsol: A newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing 19*, 1 (1998), 302–318.

[29] Pham, D., and Karaboga, D. *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks.* Springer London, Limited, 2012.

[30] Piwonski, J. *Parameter Estimates for Marine Ecosystem Models in 3-D.* PhD thesis, Faculty of Engineering, Department of Computer Science, Christian-Albrechts-Universität zu Kiel, http://macau.uni-kiel.de/receive/dissertation_diss_00016863, 2015.

[31] Piwonski, J., and Slawig, T. Metos3d: A marine ecosystem toolkit for optimization and simulation in 3-d – simulation package –. *submitted to Geoscientific Model Development Discussions* (2015).

[32] Priess, M., Piwonski, J., Koziel, S., Oschlies, A., and Slawig, T. Accelerated parameter identification in a 3D marine biogeochemical model using surrogate-based optimization. *Ocean Modelling 68*, 0 (2013), 22 – 36.

[33] Priess, M., Piwonski, J., Koziel, S., and Slawig, T. Parameter identification in climate models using surrogate-based optimization. *Journal of Computational Methods in Science and Engineering 12*, 1–2 (2012), 47–62.

[34] Rahmstorf, S., Brovkin, V., Claussen, M., and Kubatzki, C. Climber-2: A climate system model of intermediate complexity. part ii: Model sensitivity. *Climate Dynamics 17* (2001), 735–751.

[35] Roschat, C., and Slawig, T. Mathematical analysis of a marine ecosystem model with nonlinear coupling terms and non-local boundary conditions. *ArXiv e-prints*, ArXiv:1403.4461 (2014).

[36] Roschat, C., and Slawig, T. Nontrivial periodic solutions of marine ecosystem models of n-dop type. *ArXiv e-prints*, ArXiv: 1409.7540 (2014).

[37] Slawig, T., Priess, M., and Kratzenstein, C. Surrogate-based and one-shot optimization methods for pde-constrained problems with an application in climate models. In *Solving Computationally Expensive Engineering Problems*, S. Koziel, L. Leifsson, and X.-S. Yang, Eds., vol. 97 of *Springer Proceedings in Mathematics & Statistics*. Springer International Publishing, 2014, pp. 1–24.

[38] TA'ASN, S. Pseudo-time methods for constrained optimization problems governed by pde. *ICASE Report No. 95-32* (1995).

[39] TITZ, S., KUHLBRODT, T., RAHMSTORF, S., AND FEUDEL, U. On freshwater-dependent bifurcations in box models of the interhemispheric thermohaline circulation. *Tellus A 54* (2002), 89 – 98.

[40] ZICKFELD, K., SLAWIG, T., AND RAHMSTORF, S. A low-order model for the response of the atlantic thermohaline circulation to climate change. *Ocean Dynamics 54* (2004), 8–26.

Acknowledgements

I would like to thank my advisor Prof. Dr. Thomas Slawig for his advice, guidance and the support during the last years.

I am deeply indebted to my colleague Dr. Jaroslaw Piwonski for sharing his implementation expertise with me. I am thankful for the inclusion of the implementation of the One-shot strategy in his Metos-3D framework allowing a reliable and straightforward handling of the strategy in the context of generalizations, further development and applications.

Many thanks go to the groups of Prof. Dr. Andreas Griewank at Humboldt University Berlin and Prof. Dr. Nicolas R. Gauger at TU Kaiserslautern for the scientific collaboration and scientific exchange within the DFG Priority Program SPP 1253 “Optimization with partial differential Equations”. I thank the DFG for financial support.

Furthermore, I am deeply grateful to Prof. Dr. Andrea Walther and Prof. Dr. Luise Blank for organizing the workshops for developing and strengthening transferable skills. It was always a pleasure meeting you and the other women in optimization at different places in Germany.

Of course, I would like to thank all my colleagues in the working group “Algorithmic Optimal Control - CO2 Uptake of the Ocean” for accompanying me towards the completion of my work.

Finally, I would like to thank my family for their inexhausting patience and support.

Erklärung

Hiermit versichere ich,

I. dass diese Abhandlung – abgesehen von der Beratung durch den Betreuer Thomas Slawig und den Projektpartnern Andreas Griewank, Nicolas R. Gauger und Andreas Oschlies – nach Inhalt und Form meine eigene Arbeit ist,

II. dass diese Arbeit zum Teil an einer anderen Stelle bereits veröffentlicht bzw. zur Veröffentlichung eingereicht wurde, im Detail:

II.a. Simultaneous model spin-up and parameter identification with the one-shot method in a climate model example:
An International Journal of Optimization and Control: Theories & Applications (IJOCTA) 3, (2013), akzeptiert (veröffentlicht)

II.b. One-shot Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation:
Preprint SPP1253-11-03 veröffentlicht auf dem *Preprint Server des DFG-SPP1253 "Optimization with partial differential equations"*, (2008),
`http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`

II.c. Oneshot Parameter Identification - Simultaneous Model Spin-up and Parameter Optimization in a Box Model of the North Atlantic Thermohaline Circulation:
Preprint SPP1253-082 veröffentlicht auf dem *Preprint Server des DFG-SPP1253 "Optimization with partial differential equations"*, (2009),
`http://www.am.uni-erlangen.de/home/spp1253/wiki/index.php/Preprints`

II.d. Surrogate-Based and One-Shot Optimization Methods for PDE-Constrained Problems with an Application in Climate Models:
Springer Proceedings in Mathematics & Statistics, 97:1-24, Springer Verlag (2014), akzeptiert (veröffentlicht)

II.e. Optimal Design with Bounded Retardation for Problems with Non-separable Adjoints:
Leugering, G., Benner, P., Engell, S., Griewank, A., Harbrecht, H., Hinze, M., Rannacher, R., Ulbrich, S. (eds): Trends in PDE Constrained Optimization, Springer International Publishing, International Series of Numerical Mathematics Volume 165:67-84, doi: 10.1007/978-3-319-05083-6_6 (2014), akzeptiert (veröffentlicht)

II.f. Automated Extension of Fixed Point PDE Solvers for Optimal Design with Bounded Retardation:
Leugering G., Engell S., Griewank A., Hinze M., Rannacher R., Schulz V., Ulbrich M., Ulbrich S. (eds): Constrained Optimization and Optimal Control for Partial Differential Equations, Birkhäuser International Series of Numerical Mathematics 160:99122 (2012), akzeptiert (veröffentlicht)

III. dass diese Arbeit unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden ist

IV. und dass die eingereichten Veröffentlichungen einen Eigenanteil gemäß des Schreibens meines Betreuers vorweisen.

Claudia Kratzenstein
12.5.2015