

Parameter Optimization and Validation of a Marine Biogeochemical Model using a Hybrid Algorithm

J. Rückelt*, V. Sauerland*, T. Slawig*, A. Srivastav*, B. Ward†, C. Patvardhan‡

Abstract

Sensitivity computations, parameter identification and optimization for an 1-D marine biogeochemical model of *NPZD* type are presented. For the optimization a hybrid algorithm combining quantum-evolutionary and local gradient-based search methods is used. It turns out to be an efficient and flexible tool for optimization and can be easily adopted for other simulation models. For the model under investigation attainable data could be exactly identified. For realistic measurement data we argue that a certain parameter set leading to a non-optimal fit cannot be improved. Moreover we show that data uncertainty leads to a significant parameter spread. Thus we conclude that the *NPZD* model needs to be modified or extended, maybe including a modification of external forcings and/or initial conditions.

1 Introduction

In this paper we present the results of sensitivity analysis and parameter optimization for a time-dependent, spatial one-dimensional marine biogeochemical model of *NPZD* type. We investigated the dependency of the model output w.r.t. the choice of initial profiles and the length of the so-called 'spin-up' period, i.e. the time interval (in model time) that the model is run before its output is compared to given measurement data. Moreover we present the results of parameter optimization to fit given measurement or model data in a weighted least-squares sense. Here we studied both self-constructed *ideal* data that were computed by the model itself and realistic

¹Institut für Informatik, CAU Kiel. Email: joru,ts,vsa,asr@informatik.uni-kiel.de, research supported by DFG Cluster of Excellence *The Future Ocean*.

²NOC Southampton, UK

³Deemed University, Faculty of Engineering, Dayalbagh, Agra, India

observational data. For the model under scrutiny it was repeatedly shown (see e.g. [6]) that optimized parameter sets may be unreliable in the sense that widely differing sets give equally reasonable results. We want to show if this is due to the model itself or the used observational data, or, to be more specific, to their (spatial and temporal) sparsity and/or their assumed uncertainty.

For this purpose we present a powerful hybrid algorithm for parameter optimization which for the first time combines a new parallel evolutionary algorithm (EA) using quantum operators (randomized global search) and a local optimization method (deterministic local search) using Algorithmic or Automatic Differentiation (AD). The algorithm was used for the parameter optimization in this work, but it is independent of the underlying biogeochemical model and thus can serve as a tool for real parameter optimization problems in marine science.

Our main results can be summarized as follows:

1. The model results may depend on the choice of the initial profiles and on the chosen length of the spin-up.
2. The above mentioned hybrid algorithm is an efficient and useful tool for parameter optimization.
3. In our test cases we experienced that, if the data to be met are attainable (e.g. if they were computed by the model itself) and are assumed to be exact, the corresponding parameters can be uniquely identified by the optimization.
4. If the data are attainable but are assumed to be uncertain or given only with some error, the corresponding parameters can only be determined in a certain interval.
5. In our test cases the last two properties were not severely influenced by the sparsity of the data, if the latter was kept above or on the level of realistic measurement data.
6. For realistic observational data, no exact fit could be performed. Based on the above results and due to the many optimization runs with distributed starting values in the feasible parameter range, it seems quite reasonable to argue that the model is not able to reproduce the used data. Moreover it is very likely that the obtained cost function value is a global minimum. Thus a modification or extension of the *NPZD* model is needed.

We want to emphasize that especially the results 3 and 5 refer to the large number of simulation and optimization runs we performed. They are not analytically proved. It can be the case that in some other special cases and with different choices of the parameters, different results may show up.

In further research we attempt to prove mathematically the global optimality of the obtained cost function value (see point 6 above).

The structure of the paper is as follows: We describe the model, the used data and the cost function to be optimized in the next section. In the third section we describe the hybrid optimization algorithm and its main parts, namely the evolutionary algorithms and the local optimizer. Afterwards we present our results and end the paper with some concluding and summarizing remarks.

2 NPZD Model

A one-dimensional maritime biogeochemical model that simulates the interaction of dissolved inorganic nitrogen N , phytoplankton P , zooplankton Z and detritus D was developed by Schartau and Oschlies [1] in 2003, with the aim of simultaneously reproducing observations at three North Atlantic locations by the optimization of free parameters within credible limits.

The FORTRAN implementation of their model is currently used within the interdisciplinary research project "Excellence Cluster Future Ocean" at Christian-Albrechts-University, Kiel. The parameter optimization of ocean models is one of the integral parts of the excellence cluster.

Here we give a rough description of the model and its implementation.

2.1 Mathematical model

The (mmol N m^{-3}) concentrations of dissolved inorganic nitrogen, phytoplankton, zooplankton, and detritus, denoted by N , P , Z , and D , respectively, are described by the following PDE system:

$$\frac{\partial C}{\partial t} = -w_C \frac{\partial C}{\partial z} + \frac{\partial}{\partial z} \left(K_\rho \frac{\partial C}{\partial z} \right) + sms(C), \quad (1)$$

where $C \in \{N, P, Z, D\}$, K_ρ is the turbulent mixing coefficient and w_C the sinking velocity, which is nonzero only for D .

The biogeochemical source minus sink equations of the four tracers after Oschlies and Garcon [2] are given by

symbol	equation	meaning
z		depth in water column
T		temperature
V_P	$= \mu_m \cdot (C_{ref})^{cT}$	maximum growth rate of phytoplankton
u	$= \frac{N}{k_N + N}$	factor for nutrient limited growth rate of phytoplankton
$\bar{\mu}(z)$	see Appendix	light limited growth rate of phytoplankton, according to Evans and Parslow [4]
$J(\mu, u)$	$= \min(\bar{\mu}(z), V_p)$	growth rate of phytoplankton after Liebig's Law of the Minimum
$G(\epsilon, g)$	$= \frac{g\epsilon P^2}{g + \epsilon P^2}$	zooplankton grazing function

Table 1: Auxiliary variables

$$\begin{aligned}
sms(N) &= -J(\mu, u)P + \gamma_m D + \Phi_m^z Z, \\
sms(P) &= J(\mu, u)P - \Phi_m^p P - G(\epsilon, g) \cdot Z, \\
sms(Z) &= \beta G(\epsilon, g)Z - \Phi_m^z Z - \Phi_z^* Z^2, \\
sms(D) &= (1 - \beta)G(\epsilon, g)Z + \Phi_z^* Z^2 + \Phi_m^p P - \gamma_m D,
\end{aligned} \tag{2}$$

see tables 1,2.

Here, the original model was simplified by substituting the temperature dependent variables $\gamma(T) = \gamma_m \cdot (C_{ref})^{cT}$, $\Phi_z(T) = \Phi_m^z \cdot (C_{ref})^{cT}$, $\Phi_p(T) = \Phi_m^p \cdot (C_{ref})^{cT}$ by constants, again named $\gamma_m, \Phi_m^z, \Phi_m^p$, respectively.

2.2 Data

Assimilated data are from the Bermuda Atlantic Time-Series Study (BATS; 31N 64W).

Measured ecological BATS data corresponding to the biogeochemical model are available for dissolved inorganic nitrogen (DIN) (mmol N m^{-3}), chlorophyll a (Chl a) (mg (Chl a) m^{-3}), zooplankton biomass (ZOO) (mmol N m^{-3}), particulate organic nitrogen (PON) (mmol N m^{-3}), and carbon fixation (or primary production as carbon uptake, CPP) ($\text{mmol C m}^{-3} \text{ day}^{-1}$). Except for zooplankton biomass, the tracer value is its concentration at the corresponding depth. For zooplankton biomass, the tracer value is the integrated concentration in the water column from the given depth (approximately 200 meters) up to the ocean surface (zooplank-

parameter index	symbol	value	unit	
	C_{ref}	1.066	1	growth coefficient
	c	1	$^{\circ}\text{C}^{-1}$	growth coefficient
	R	6.625	1	molar carbon to nitrogen ratio
	k_w	25	m^{-1}	PAR extinction length
	f_{PAR}	0.43	1	short-wave PAR fraction
1	β	$[0, 1]$	1	assimilation efficiency of zooplankton
2	μ_m	\mathbb{R}_0^+	d^{-1}	phytoplankton growth rate parameter
3	α	\mathbb{R}_0^+	$\text{m}^2\text{W}^{-1}\text{d}^{-1}$	slope of photosynthesis vs light intensity
4	Φ_m^z	\mathbb{R}_0^+	d^{-1}	zooplankton loss rate parameter
5	κ	\mathbb{R}_0^+	$\text{m}^2(\text{mmol N})^{-1}$	light attenuation by phytoplankton
6	ϵ	\mathbb{R}_0^+	$\text{m}^6(\text{mmol N})^{-2}\text{d}^{-1}$	grazing encounter rate
7	g	\mathbb{R}_0^+	d^{-1}	maximum grazing rate
8	Φ_m^p	\mathbb{R}_0^+	d^{-1}	phytoplankton linear mortality
9	Φ_z^*	\mathbb{R}_0^+	$\text{m}^3(\text{mmol N})^{-1}\text{d}^{-1}$	zooplankton quadratic mortality
10	γ_m	\mathbb{R}_0^+	d^{-1}	detritus remineralization rate
11	k_N	\mathbb{R}_0^+	mmol N m^{-3}	half saturation for NO_3 uptake
12	w_s	\mathbb{R}_0^+	m d^{-1}	detritus sinking velocity
13	Φ_m^*	\mathbb{R}_0^+	$\text{m}^3(\text{mmol N})^{-1}\text{d}^{-1}$	phytoplankton quadratic mortality

Table 2: Model parameters. Parameters with a number in the first column were optimized.

ton is measured by dragging a 200 μ m net to the surface).

The state variable N corresponds to DIN, P to Chl, Z to ZOO, $P + Z + D$ to PON and $PP = J(\mu, u) \cdot P \cdot R$ to CPP. The zooplankton conversion is from mesozooplankton to total zooplankton. The original data came from [7]. The observed zooplankton ZOO is transformed to $Z_{obs} = 1.23 \cdot ZOO + 0.097$, see also [6].

2.2.1 Initial profiles

An initial vertical concentration profile for N is calculated as the mean depth profile derived from the DIN observations, the components of the other profiles (P, Z, D, PP) are set following [3].

2.3 Forcing

The BGC model is forced by output from the OCCAM global circulation model (<http://www.noc.soton.ac.uk/JRD/OCCAM/>). This output estimates the hourly vertical profiles of temperature T (in $^{\circ}C$) and vertical diffusivity w_c (in m^2s^{-1}), respectively.

2.4 Integration

The time resolution is one hour, the vertical grid consists of 66 layers with thickness increasing with depth. The source minus sink equations (2) are integrated using a Eulerian scheme four times per hour after which the drift and diffusion equation (1) is solved.

2.5 Cost function

Let $j \in \{N, P, Z, D, PP\}$ be a tracer, a be a runtime year and $\mathcal{N}(j, a)$ denote the number of measurements of tracer j in the year a . For any $i \in \mathcal{N}(j, a)$ the i -th measurement of tracer j in the year a is written as $y(j, a, i)$ and the associated time and depth as $t(j, a, i)$ and $z(j, a, i)$, respectively.

The depth of the middle of the k -th model layer, $1 \leq k \leq 66$, is denoted by $z_m(k)$. We further set $z_m(0) = 0$. The nearest upper and lower layers to $z(j, a, i)$ (with respect to their centers) are determined as

$$\begin{aligned} \bar{k}_{j,a,i} &= \max\{0 \leq k \leq 66 \mid z_m(k) \leq z(j, a, i)\} \\ \underline{k}_{j,a,i} &= \min\{1 \leq k \leq 66 \mid z_m(k) \geq z(j, a, i)\} \end{aligned}$$

We define $\tilde{f}(PP, a, h, k)$ as the modeled carbon-primary production in the 24 hours time interval $[h - 12, h + 11]$ around the h -th hour of the year a in

layer k , whereas $\tilde{f}(j, a, h, k)$ is only the associated single model output value, if $j \neq PP$. We further set $\tilde{f}(j, a, h, 0) = \tilde{f}(j, a, h, 1)$ for every j, a and h .

Now, for every measurement $y(j, a, i)$ a corresponding value $f(j, a, i)$ can be derived from the model output. Both values are scaled with respect to the same measuring unit. For $j \neq Z$, the linear interpolation

$$f(j, a, i) = \alpha \cdot \tilde{f}(j, a, h_{j,a,i}, \bar{k}_{j,a,i}) + (1 - \alpha) \cdot \tilde{f}(j, a, h_{j,a,i}, \underline{k}_{j,a,i}),$$

with $h_{j,a,i} = \lfloor t(j, a, i) \rfloor$ and

$$\alpha = \frac{z_m(k_2) - z(j, a, i)}{z_m(k_2) - z_m(k_1)}$$

is taken, whereas $f(Z, a, i)$ is $\frac{1}{z(j,a,i)}$ times the integral of the accordant piecewise linear function over the water column $[0, z(j, a, i)]$.

Now, the over all cost function is calculated as

$$F = \frac{1}{\nu} \sum_{a \in A} \sum_{j=1}^5 \frac{1}{\sigma_j^2 \cdot \mathcal{N}(j, a)} \sum_{i=1}^{\mathcal{N}(j,a)} (f(j, a, i) - y(j, a, i))^2, \quad (3)$$

where A is the set of model runtime years except for the first year (spin-up year), ν is the cardinality of $\{(j, a) \mid 1 \leq j \leq 5, a \in A, \mathcal{N}(j, a) \neq 0\}$ and σ weights the different tracers and is currently set to 0.1 for N , 0.01 for P , 0.01 for Z , 0.0357 for D and 0.025 for PP .

3 Optimization

In this section we formalize the minimization problem we studied and give some details on the used software.

The optimization problem is of least-squares type with box constraints and can be written as

$$\min_{\mathbf{x} \in X_{ad}} F(\mathbf{x}) := \|\mathbf{y}(\mathbf{x}) - \mathbf{y}_d\|^2 \quad s.t. \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \quad f : \mathbb{R}^N \mapsto \mathbb{R}, \quad (4)$$

i.e. the minimization of the composition of the $NPZD$ model and the cost function w.r.t. the parameters, summarized in the vector $\mathbf{x} \in X_{ad} \subset \mathbb{R}^n$. The norm in F is a weighted Euclidean norm. The vector \mathbf{y} denotes the discrete values of the four tracers N, P, Z, D on the space-time grid supplied by the model, and \mathbf{y}_d are given data. Usually the model produces output on much more discrete grid points than data are available, so in fact \mathbf{y} in the cost function first has to be restricted (and maybe also interpolated)

to the grid points were the data \mathbf{y}_d are given. For simplicity we omit the restriction and/or interpolation operator here. The parameters are given in table 2. The vectors $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ represent the parameter bounds, where also $\mathbf{l}, \mathbf{u} \in (\mathbb{R} \cup \{\pm\infty\})^n$ is possible.

To this end we combined two optimization strategies, namely a evolutionary algorithm (or randomized direct search) and a sequential quadratic programming (SQP) algorithm as a gradient based one. Gradient based algorithms are supposed to find local minima but to also get stuck there, whereas evolutionary algorithms are expected to search more globally, having a hard time to actually compute a (global) minimum to a given precision. In the next two subsections we describe this two methods, thereafter we turn to our hybrid algorithm which combines the two in a very promising and flexible way.

3.1 Quantum Evolutionary Computing

We briefly describe the framework of standard evolutionary computing and give an overview of our enhanced quantum evolutionary algorithm.

3.1.1 Evolutionary Algorithm

One iteration of an evolutionary algorithm is referred to as a generation, one parameter set to be tested as an individual, the parameters as genes and a set of parameter sets as a population of size N_2 . Figure 1 is a row schematic of the basic principles of a classical evolutionary algorithm. The EA framework we use works as follows: After using a randomized method to choose an initial population, in each iteration a best ('fittest') individual is determined (lowest costfunction, $N_3 = 1$) and tried to improve upon. After that a so called crossover operator (BLX- α) tries to improve upon each individual of the population.

The mutation operator firstly perturbs a subset of the parameters by random jumps. After perturbation, the new parameter set is optionally sent to an Armijo-type search routine. By that we mean a line search which yields a 'large' stepsize. The direction is that of a randomly chosen parameter. Perturbations generally improve the search space exploration, whereas the Armijo-type search provides faster convergence and higher accuracy but also higher probability to get stuck in local optima.

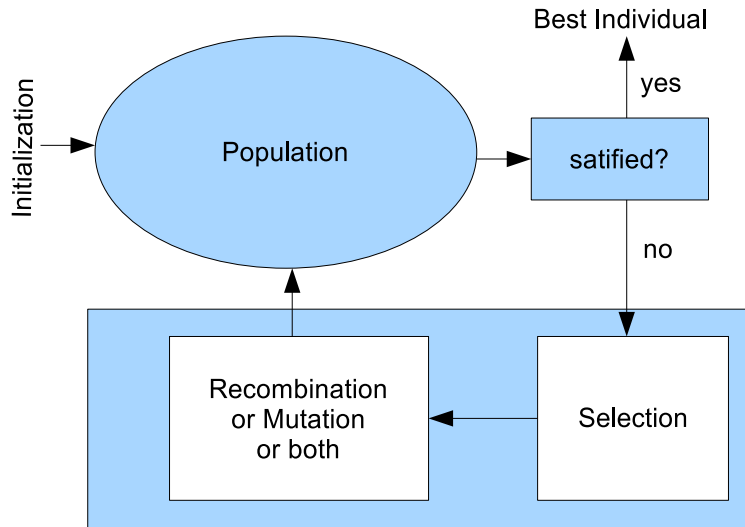


Figure 1: schematic of a single population EA (according to 3.(a) – 3.(c) of the hybrid outline)

3.1.2 Real-parameter QEA (RQEA)

A quantum evolutionary algorithm (QEA) is a population based probabilistic evolutionary algorithm that integrates concepts from quantum computing for higher representation power and robust search. It maintains a population of individuals in so called quantum bits or qubits, which are pairs of complex numbers $(\alpha, \beta)^T$ that satisfy $|\alpha|^2 + |\beta|^2 = 1$ representing probabilities to generate corresponding solution vectors of a given problem: QEAs usually deal with binary problems using the concept of observation (randomized rounding) to generate candidate binary solutions from qubit strings. RQEAs variate the concept of quantum evolutionary algorithms (QEAs) [10] to be applicable to problems with continuous parameter domains. One has to use adapted quantum evolution operators to generate candidate solution strings that comprise real parameters. The idea introduced in [9] is to maintain pairs of qubit strings and real solution strings as individuals. There are essentially two operations for the evolution of the population:

a) Update of Qubits - Quantum Gates A quantum gate acting on \mathbb{C}^n is a unitary matrix $C \in \mathbb{C}^{n \times n}$. We use 2-dimensional rotation gates, which are 2×2 -matrices representing rotation in \mathbb{C}^2 , where

$$C_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix},$$

and θ is the rotation angle.

Suppose we were given a qubit $q = (\alpha, \beta)$ and a real parameter x . Suppose further that the fitness value of x is good. In the next step, where we want to generate new individuals, we would like to generate parameters, which are close to the good one x on the one hand, but which are also diverse on the other hand. We capture both aims by updating the qubit $q = (\alpha, \beta)$ to a qubit $q' = (\alpha', \beta')$ which is close to q . This is mathematically accomplished by rotating q with a small angle θ , so

$$q' = \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = C_\theta \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = C_\theta q.$$

In the next step, individuals according to q' are generated. This process generalizes to n dimensions in a natural way. Given a n -dimensional qubit

$$Q = \begin{pmatrix} \alpha_1 & \cdots & \alpha_n \\ \beta_1 & \cdots & \beta_n \end{pmatrix},$$

we choose rotation matrices $C_{\theta_1}, \dots, C_{\theta_n}$, and update Q to

$$Q' = \begin{pmatrix} \alpha'_1 & \cdots & \alpha'_n \\ \beta'_1 & \cdots & \beta'_n \end{pmatrix},$$

where

$$\begin{pmatrix} \alpha'_i \\ \beta'_i \end{pmatrix} = C_{\theta_i} \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix}$$

for all i . With $p_i = |\alpha'_i|^2$, $p = (p_1, \dots, p_n)$ gives relative step sizes from a given solution toward a selected solution.

In evolutionary computing, the angles θ_i are crucial and have to be chosen in a careful way.

b) Evolution Operators RQEA generates candidate states during the search process in the following way. A population of N_p n -dimensional qubits, Q_t^i , $i = 1, 2, \dots, N_p$, is maintained in the t^{th} iteration where $N_p = N_1 \cdot N_2$ is the population size chosen as per requirement. Higher values of N_p enable a wider exploration of the search space with correspondingly greater computational effort. From each of the N_p n -dimensional qubits, N_o offspring qubits $P_t^{(i,j)}$, $j = 1, 2, \dots, N_o$, are generated using the rotation gate operator. The rotation operation is done such that the probability distributions $p_t^{(i,j)}$ according to the offspring qubits $P_t^{(i,j)}$ provide higher probability to generate either the best so far real-parameter solution (neighborhood operator 1 (N01)) or a solution with parameters that are closer to its lower or upper constraint

(neighborhood operator 2 (N02)). Each $P_t^{(i,j)}$ is transformed to solution space by setting

$$y^{(i,j,l)} = |\alpha^{(i,j,l)}|^2 (y_{\max} - y_{\min}) + y_{\min}, \quad l \in [n]$$

with y_{\min} , y_{\max} being the box constraints of the l^{th} parameter (N02) or the minimum (maximum) of $x_t^{(i,j,l)}$ and the accordant gene in the very best solution, x_{best}^l (N01). The fitness of each of these is determined and the best of them is identified according to the fitness criterion. Let the qubit $P_t^{(i,k)}$ be the one from which the best solution was obtained. If this solution is better than the solution obtained from Q_t^i , $P_t^{(i,k)}$ replaces Q_t^i to become Q_{t+1}^i and $y^{(i,k)}$ replaces x_t^i to become x_{t+1}^i .

3.1.3 Inherent Parallelization

As well as a standard EA the RQEA can easily be parallized using subpopulations as described in the outline of the hybrid algorithm.

3.2 Local optimization method

For the local optimization we used the software CFSQP, see [12], which is based on sequential quadratic programming (SQP) and is well suited to the problem on hand. It can moreover treat general nonlinear constraints. It was our intention to use an off-the-shelf local optimization routine and *not* one that was specifically designed for the model under investigation. The above mentioned code can be easily replaced by another local optimizer that takes into account the box constraints. It was already mentioned in [6] that a local method that does not can lead to infeasible (i.e. negative) parameters and/or model output for N , P , Z or D .

If the coded mathematical function of interest is smooth enough, efficient code for exact derivatives can be generated automatically by the technology of *Algorithmic or Automatic Differentiation* (AD), see e.g. [5]. This was the case here, the only non differentiable term being $J(\mu, u)$, see table 1. We used the proprietary AD software TAF, see [11], in the forward mode. The combined effort of a function evaluation and its 12-dimensional gradient was roughly 5 times that of a function evaluation alone. A finite differences approximation to the gradient would cost at least 13 function evaluations. Moreover, the use of AD-generated gradients avoids approximation errors and instabilities which are well-known for numerically approximated derivatives.

3.3 Hybrid optimization algorithm

The general setting of a hybrid optimization method can be summarized in the following

Hybrid evolutionary-deterministic algorithm:

1. Choose the number N_1 of available or desired processors (if the method is used in parallel version).
2. Choose randomly a population of $N_1 \cdot N_2$ individuals, i.e. a set of different parameter vectors $\mathbf{x}^{(ij)} \in X_{ad}, i = 1, \dots, N_1, j = 1, \dots, N_2$, and distribute them among the processors.
3. On each processor $i = 1, \dots, N_1$:
 - (a) Evaluate the cost function for every individual, i.e. compute $F(\mathbf{x}^{(ij)})$ for all corresponding $\mathbf{x}^{(ij)}, j = 1, \dots, N_2$.
 - (b) Choose the $N_3 \leq N_2$ best individuals, i.e. those with the lowest cost function value.
 - (c) For each of these best individuals $\mathbf{x}^{(ijk)}, k = 1, \dots, N_3$:
 - i. Do N_4 times repeatedly:
 - A. Perform a mutation step, i.e. change randomly some components of the parameter vector $\mathbf{x}^{(ijk)}$ to obtain $\tilde{\mathbf{x}}^{(ijk)}$.
 - B. Perform a cross-over step, i.e. combine the parameters of two individuals $\tilde{\mathbf{x}}^{(ijk)}, \tilde{\mathbf{x}}^{(ijk')}, k, k' \in \{1, \dots, N_3\}$, to get new ones $\bar{\mathbf{x}}^{(ijk)}, \bar{\mathbf{x}}^{(ijk')}$.
 - C. If a 'new' individual $\bar{\mathbf{x}}^{(ijk)}$ is better than the original one (in this iteration), $\mathbf{x}^{(ijk)}$, replace the old by the new one, i.e. set $\mathbf{x}^{(ijk)} := \bar{\mathbf{x}}^{(ijk)}$.
 - ii. Compute N_5 steps of a local optimization method to obtain individuals $\hat{\mathbf{x}}^{(ijk)}$.
 - (d) Determine the best individual

$$\mathbf{x}^{(ij^*)} := \arg \min_j F(\mathbf{x}^{(ij)})$$

on the processor.

4. (a) Gather the best individuals from all processors, i.e. determine

$$\hat{\mathbf{x}} = \arg \min_i F(\mathbf{x}^{(ij^*)}).$$

(b) Distribute it among all processors to replace the best, i.e. set

$$\mathbf{x}^{(ij^*)} := \hat{\mathbf{x}} \quad \text{for all } i = 1, \dots, N_1.$$

5. If no improvement in F or no significant change in $\hat{\mathbf{x}}$ is made, stop.
Else: go to 3.

The algorithm has several parameters that allow to steer it either into a more (or purely) stochastic or deterministic direction:

- (1) Omitting the local method ($N_5 = 0$) in step 3(c)ii gives a quantum-evolutionary algorithm, in this case with a mutation operator that includes a line search.
- (2) On the other hand a local optimization method with randomly distributed starting values is obtained when omitting the evolutionary-type mutation and cross-over operations, or, more formally, setting $N_2 = N_3 = 1, N_4 = 0$, and N_5 large. Steps 3d and 4b are obsolete then.

Clearly a single-processor version is obtained for $N_1 = 1$, and the order of the evolutionary and the local minimizations theoretically can be exchanged (steps 3(c)i and 3(c)ii).

3.4 Implementation and experiments

The hybrid algorithm was essentially implemented as described in the former subsections. But our implementation of the mutation operator contains a very simple realization of the 'neighbourhood operators' described in subsection 3.1.2. A full implementation of the RQEA using qubits is in work.

Using MPI, the implementation was parallized on the linux cluster of the computing center of the Christian-Albrecht-University of Kiel (CAU). The used batch class is reserved for researchers of the Cluster of Excellence *The Future Ocean* and consists of 24 8-core barcelona nodes. For the experiments with the hybrid algorithm dominated by the evolutionary operators (version (1) avbove) and a simple coordinate-oriented line search, we used $N_1 = 64$ processes, each process dealing with one subpopulation of size $N_2 = 11$, migration of clones of the very best individual to each population being performed in each generation (according to 4. in the outline). For the version which was dominated by the gradient-based local SQP method (version (2) above) we in most cases used 16 processors, and on each of them a stochastically generated starting value was used for the optimization. Thus also in this version the algorithm is easy parallizable.

4 Model sensitivity with respect to initial profile and spin-up

In this section we present sensitivity computations of the model output and cost function w.r.t initial profiles and the choice and duration of the spin-up.

A typically run of the optimization is performed as follows. After deciding on initial profiles for the tracers N, P, Z and D , a so-called 'spin-up year' is performed and then the model output of the years 1991-1995 is evaluated against model-generated ideal or observational data via the cost function as explained in the first section. Only the upper 20 depth levels (the euphotic zone) are evaluated, and so only the upper 32 levels (of the total 66) of the model are used. Using more levels doesn't affect the result (using the cost function F as indicator) but increases the computational cost.

4.1 Dependency on the initial N -profile

Technically, the spin-up year is a model year the output of which does not enter into the cost function F . The term spin-up implies, that after that period the state is 'independent' of the initial state, i.e. the differences for different initial values are neglectable.

As a first point of validation, we checked upon that. We take

- some fixed initial profile for N – meaning to represent an 'initial NO₃ inventory' – for the spin-up (the other initial profiles are always kept constant in the model),
- and a fixed parameter vector, called 'target', (same as for reconstruction test, see next section).

For these we used 1990 as spin-up year and computed the model output \mathbf{y} for the years '91-'98. This output was used as data \mathbf{y}_d . Now other initial profiles were used and the cost function was calculated. The other three initial profiles stayed fixed.

We now replaced the original profile

1. by the mean of the spin-up year 1990. The result (see table 3), shows a difference in the cost in all following years, compared to the zero cost function value when the original spin-up profile was used.
2. and then by the mean of January of the spin-up year 1990,

Thus model output and the cost function are *not* independent of the initial profile. The different profiles can be seen in figure 2.

initial profile	contribution to cost F in year									
	'90	'91	'92	'93	'94	'95	'96	'97	'98	all
original	*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
'90 mean	*	7.79	6.77	1.72	2.41	3.26	1.73	1.19	0.49	3.17
Jan '90 mean	*	0.117	0.108	0.033	0.045	0.059	0.036	0.025	0.013	0.055

Table 3: Initial N -Profile: Yearly and overall cost using one-year spin-up. Model output used as data. Zero if original profile used (first row), * = spin-up year. In the last column 'all' means all years after the spin-up. Cost function values are weighted Euclidean norms, thus the sum over the years does not equal the total cost over all years.

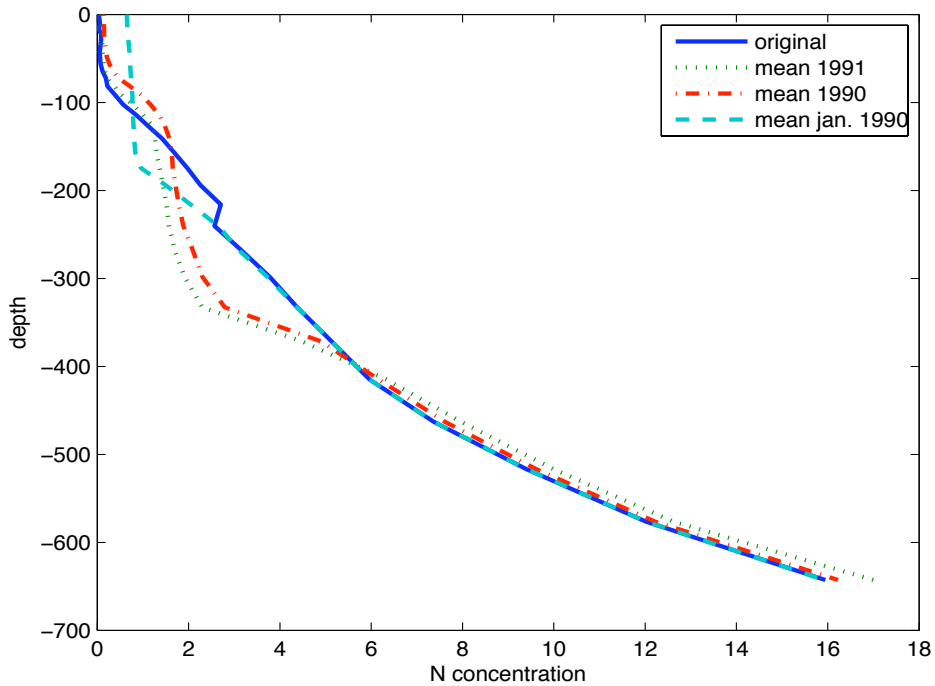


Figure 2: Initial N profiles

4.2 Additional dependency on the choice of the spin-up year

Secondly, we tested the dependency of the model output for the different choices of the initial profiles as in the last subsection, but now additionally for different spin-up years. Ideally the model should reproduce the same output after the spin-up whatever year is taken as for the spin-up year. Tables 4, 5, and 6 show the results when taking either

1. the original profile (see table 4)
2. the mean of the corresponding spin-up year (see table 5),
3. and again as in the last subsection the mean of January of the corresponding spin-up year (table 6).

Choice 1 can obviously lead to large errors. Choice 2 seems to improve the situation, and choice 3 mostly gives good accuracy.

	contribution to cost F in year									
initial profile	'90	'91	'92	'93	'94	'95	'96	'97	'98	all
original	*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
original	*	7.79	6.77	1.72	2.41	3.26	1.73	1.19	0.49	3.17
original		*	8.25	1.83	2.58	3.48	1.84	1.27	0.52	2.83
original			*	5.97	8.79	11.16	6.27	4.74	1.86	6.47
original				*	9.43	9.94	3.21	1.76	0.68	5.00
original					*	15.71	1.45	0.21	0.07	4.36
original						*	5.04	2.00	0.47	2.50
original							*	6.40	1.54	3.97
original								*	2.72	2.72

Table 4: As table 3, but now the spin-up and thus the evaluation of the cost was started at different times. Here the original initial profile that produces $F = 0$ (see first row), when 1990 was taken as spin-up year, was used.

As a consequence one could argue that one spin-up year is too short, at least to generate model output that is independent from initial profile and the choice of the spin-up year. Our guess based on tables 4, 5, and 6 would be, that the required time, depending on the required accuracy, should be well above eight years. A good estimate for the initial profiles would probably be more worthwhile. For this end, the model could be started at a point with sufficient observational data in a neighborhood.

	contribution to cost F in year									
initial profile	'90	'91	'92	'93	'94	'95	'96	'97	'98	all
original	*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
mean '90		*	0.364	0.072	0.099	0.142	0.111	0.103	0.070	0.138
mean '91			*	0.105	0.141	0.197	0.140	0.121	0.076	0.130
mean '92				*	1.380	1.710	0.580	0.280	0.104	0.810
mean '93					*	0.280	0.770	0.670	0.23	0.490
mean '94						*	0.364	0.405	0.141	0.305
mean '95							*	0.454	0.121	0.288
mean '96								*	0.261	0.261

Table 5: As table 4, but mean of previous year taken as initial profile.

	contribution to cost F in year									
initial profile: mean of	'90	'91	'92	'93	'94	'95	'96	'97	'98	all
Jan '90	*	0.117	0.108	0.033	0.045	0.059	0.036	0.025	0.013	0.055
Jan '91		*	0.394	0.097	0.131	0.179	0.107	0.076	0.038	0.146
Jan '92			*	0.089	0.117	0.157	0.089	0.060	0.028	0.090
Jan '93				*	1.590	1.585	0.516	0.266	0.107	0.813
Jan '94					*	0.169	0.025	0.009	0.004	0.051
Jan '95						*	0.001	0.000	0.000	0.004
Jan '96							*	0.118	0.021	0.070
Jan '97								*	0.038	0.038

Table 6: As table 4, but mean of January taken as initial profile.

Incidentally, our 'best fit' (see respective section) is obtained using interpolated observational data of January,15,1991.

4.3 Choosing longer spin-up periods

A common way to spin up a model is to repeat one year with fixed forcing and parameters until a steady stationary or periodic model output is achieved. We performed this extended spin-up to check if in this case the dependency w.r.t to the initial profile changes.

After roughly 50 repetitions of one year a fixed point (i.e. a periodic solution) is found. As can be seen in table 7, this fixed point is not independent of the initial profile and does not necessarily yield a low cost F , table 7. The

initial profile	spin-up period	contribution to cost F in year								
		'91	'92	'93	'94	'95	'96	'97	'98	all
Jan '90	2x '90	6.15	5.95	1.51	2.10	2.81	1.38	0.87	0.30	2.60
Jan '90	6x '90	73.7	51.3	16.8	24.7	26.6	14.2	10.2	2.9	27.6
Jan '90	20x '90	126.2	94.2	45.2	51.0	53.0	32.5	27.3	7.1	54.6
Jan '90	50x '90	132.5	103.6	50.5	54.6	57.1	35.9	30.8	7.9	59.1
Jan '90	100x '90	132.7	103.9	50.6	54.7	57.3	36.0	30.9	7.9	59.3
'90	100x '90	134.2	107.5	52.8	56.1	59.0	38.1	33.5	8.9	61.3
'91	100x '90	134.2	107.3	52.7	56.1	59.0	38.0	33.4	8.9	61.1
'92	100x '90	134.8	109.0	53.6	56.7	59.7	38.9	34.5	9.3	62.1
'94	100x '90	133.3	105.3	51.5	55.3	58.0	36.8	31.9	8.3	60.0
orig.	50x '90	132.1	102.7	49.9	54.2	56.6	35.3	30.1	7.6	58.6
orig.	100x '90	132.3	103.0	50.0	54.3	56.8	35.4	30.2	7.7	58.7

Table 7: Repeated spin-up (over year 1990) with different initial N -profiles: Mean of January, 1990, mean of '90 and profile used originally.

spin-up	contribution to cost F in year								
	'91	'92	'93	'94	'95	'96	'97	'98	all
'90-'98	70.70	77.86	46.76	53.03	56.29	40.58	39.68	12.56	49.68
5x '90-'98	118.2	129.1	81.5	75.1	81.6	79.0	72.2	21.4	82.3
10x '90-'98	118.3	129.3	81.6	75.2	81.7	79.2	72.4	21.5	82.4

Table 8: Repeated spin-up over longer time period (1990-1998) with so far best initial N -profile (mean of January, 1990).

same goes for longer spin-up periods, e.g. '90-'98, see table 8. Note that some coefficients dependent on the actual date and year (temperature and mixing coefficient). Therefore a spin-up '90-'98 is different from one using e.g. 9x'90 etc.

5 Optimization results

In this section we present the results of parameter optimization, first for ideal, model-generated data, and then for realistic observational data. In both cases we investigate the obtained parameter spread, i.e. the intervals of parameters that lead to similar values of the cost (up to a given tolerance).

5.1 Parameter identification

We applied the hybrid optimization algorithm on the following parameter identification or reconstruction problem:

We used the parameter vector given in [6] as default parameters, i.e. using these parameters, model output $\mathbf{y} =: \mathbf{y}_d$ for the years 1991-1995 at the times and locations of the 2469 actual observations was generated.

Then we applied the optimization using the parameter bounds $[\mathbf{l}, \mathbf{u}]$ (to be understood component-wise for the parameters in \mathbf{x}) of table 9. The initial profiles and the forcings are fixed. Thus a successful optimization should give $F \approx 0$. The interesting question is if the original parameters corresponding to \mathbf{y}_d can be obtained.

The hybrid algorithm at its one extremum, i.e. just using stochastically chosen initial values for the parameters and then taking the local optimizer CFSQP with AD generated gradients runs up a cost of roughly 600 – 1000 function evaluations per optimization. For the chosen target parameter vector about a third of all optimizations starting from randomly chosen parameter vectors (with a value of the cost function at the order of $F \approx 10^2$) succeeded in reconstructing the target parameter vector with $F \approx 10^{-6}$. These numbers were also obtained for other choices of targets, but of course it can not be guaranteed that they can not be worse for special targets. The other trials terminate in local minima or due to 'failure' of the algorithm.

Dealing with real world data in the same way, best data-fits are also found in about a third of the optimizations, which gives rise to the assumption that the found local optima are global. Evolutionary algorithms are generally more costly but have higher search space exploration abilities. The computational performance of our hybrid algorithm was best when it was applied in the form that no mutation and cross-over operators were used, but the local SQP method was started from equally-distributed randomly chosen starting points. The performance of this variant during a typical successful reconstruction is seen in figure 3. This and the many experiments with the hybrid algorithm run in a more 'evolutionary- dominated' variation with several populations and just a coordinate-wise (also known as Hooke-Jeeves method) local minimization in every step support our assumption of global

optimality.

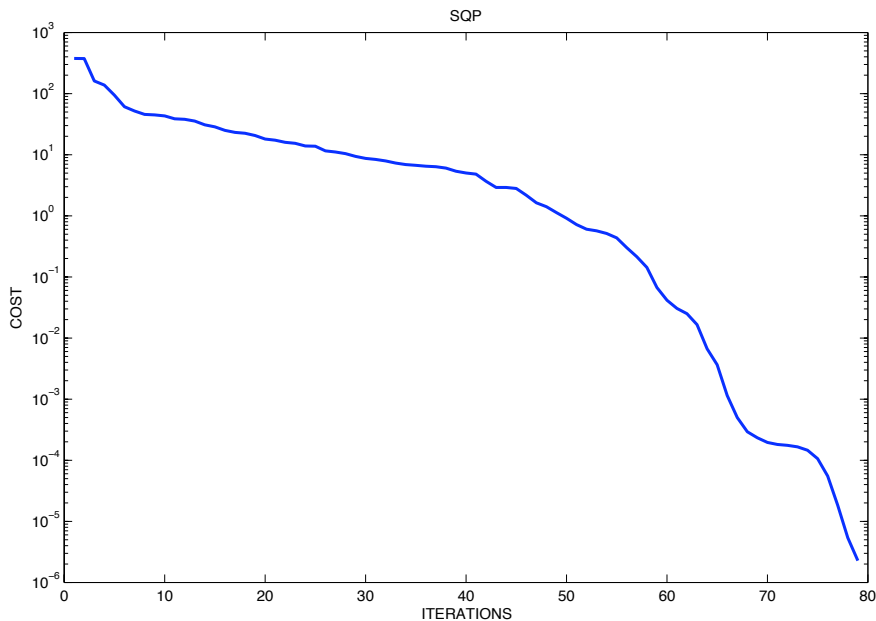


Figure 3: Typical successful optimization with CFSQP.

5.2 Parameter Spread

In this section we show how the optimized parameter values depend on a given accuracy of the cost function F . Due to observational error or uncertainty, parameter vectors yielding a cost above a found minimum for a given set of observations are considered as equally valid, see [6],[8]. For the *NPZD* model and the used data set, we used a threshold of $\Delta F = 1.165$, see [6].

We first investigated the dependency on the optimized parameters our algorithm gives when the stopping criterion was released, i.e. the threshold ΔF was increased. For the ideal data taken from a model run with given target parameter vector \mathbf{x}_d , we expect that the value $F < \Delta F$ can be reached for small ΔF .

We first show how the range or spread of parameters is for a 'realistic' value of ΔF . When the threshold ΔF suggested for the observational data is used, the parameter spread is quite big, see table 9. The parameters with $F \leq \Delta F = 1.165$ found in the $[min, max]$ ranges are not necessarily local minima, and ranges might prove wider after a more thorough search.

					$\Delta F = 1.165$			
i	\mathbf{x}_i	\mathbf{x}_{di}	\mathbf{l}_i	\mathbf{u}_i	$\min(\mathbf{x}_i)$	$\max(\mathbf{x}_i)$	%	outlier $F = 121.9$
1	β	0.7500	0.0	1.0	0.6764	0.9995	33.3	0.7498
2	μ_m	0.6000	0.01	1.46	0.5437	0.9423	57.1	0.81
3	α	0.0250	0.0	0.253	0.0206	0.0340	35.9	0.0242
4	Φ_m^z	0.0100	0.0	0.63	0.0005	0.0170	94.5	0.0100
5	κ	0.0300	0.0	0.073	0.0115	0.0332	61.5	0.0145
6	ϵ	1.0000	0.01	4.0	0.8560	2.0000	100.0	0.9747
7	g	2.0000	0.01	4.0	1.4370	3.9660	98.3	2.0000
8	Φ_m^p	0.0100	0.001	0.63	0.0015	0.0119	84.9	0.0111
9	Φ_z^*	0.2050	0.0	1.0	0.0354	0.2640	82.7	0.2460
10	γ_m	0.0200	0.0	0.15	0.0001	0.0225	99.4	0.0010
11	k_N	0.5000	0.01	1.0	0.4487	0.7623	52.5	0.4554
12	w_s	6.0000	0.1	128.0	2.8650	6.6180	52.3	5.7490

Table 9: Parameter spread: Reconstruction of target parameter vector \mathbf{x}_d in bounds $[\mathbf{l}, \mathbf{u}]$ (component-wise) from model output $\mathbf{y}_d = \mathbf{y}(\mathbf{x}_d)$ at same points (2469) as observed data. For $\mathbf{x}_i \in [\min, \max]$ sets with cost below $\Delta F = 1.165$, the value also suggested in [6], are found. % is maximal found deviation from target parameter \mathbf{x}_{di} in percent. Outlier is a set found in $[\min, \max]$.

5.3 Dependency on data sparsity

To see if too sparse observations might cause the problem, we repeat the reconstruction with model output for the first hour of every day for the years 1991-1995 for the upper twenty spatial grid points for N, P, D, PP and one per hour for Z (177025 points) and obtained the same parameter vector as before. A spread of the same magnitude is found, as seen in the left half of table 10.

5.4 Parameter spread with respect to data uncertainties

To lower the error in the parameters, the accepted accuracy in the cost must be tightened, see tables 10,11. Figure 5 shows the trajectories of the target parameter vector and some other parameter vector giving a cost within the accepted margin ($F = 1.04$).

The parameter spread is significantly reduced when taking ΔF even smaller, see table 11. The value of $\Delta F = 10^{-2}$ used there correspond to

			$\Delta F = 1.165$				$\Delta F = 0.5$			
i	\mathbf{x}_i	\mathbf{x}_{di}	$\min(\mathbf{x}_i)$	$\max(\mathbf{x}_i)$	%	outlier $F = 112.4$	$\min(\mathbf{x}_i)$	$\max(\mathbf{x}_i)$	%	outlier $F = 73.5$
1	β	0.750	0.4924	1.0000	34.3	0.4924	0.5000	0.9609	33.3	0.50
2	μ_m	0.600	0.4694	1.0240	70.7	1.0240	0.4953	1.0100	68.3	1.0100
3	α	0.025	0.0181	0.0360	43.9	0.0360	0.0201	0.0310	23.8	0.0310
4	Φ_m^z	0.010	0.0044	0.0166	66.3	0.0166	0.0067	0.0141	40.7	0.0141
5	κ	0.030	0.0139	0.0522	74.1	0.0139	0.0226	0.0423	41.0	0.0226
6	ϵ	1.000	0.8397	1.4000	40.0	0.8397	0.8649	1.2260	22.6	0.8649
7	g	2.000	0.8994	4.0000	100.0	1.705	1.2220	3.8060	90.3	1.7050
8	Φ_m^p	0.010	0.0050	0.0144	50.0	0.0050	0.0079	0.0124	24.4	0.0079
9	Φ_z^*	0.205	0.1000	0.4000	95.1	0.4000	0.1534	0.2748	34.0	0.2748
10	γ_m	0.020	0.0127	0.0660	230.0	0.0127	0.0130	0.0333	66.5	0.0130
11	k_N	0.500	0.2286	1.0000	100.0	1.0000	0.2722	1.0000	100.0	0.2722
12	w_s	6.000	3.8320	20.4700	241.2	6.214	4.0000	10.0700	67.8	6.2140

Table 10: As table 9, but now for denser data. Model output at 177025 points used as desired state \mathbf{y}_d .

a relative reduction of the cost of about four orders of magnitude, since a typical value for the starting point of an optimization is around $F \approx 200$. The results shows that, assuming that the data are given with little error or uncertainty, a successful parameter identification can be performed using the applied hybrid optimization method.

A visualization of the parameter spread for the different ranges of ΔF for one example can be seen in figure 4.

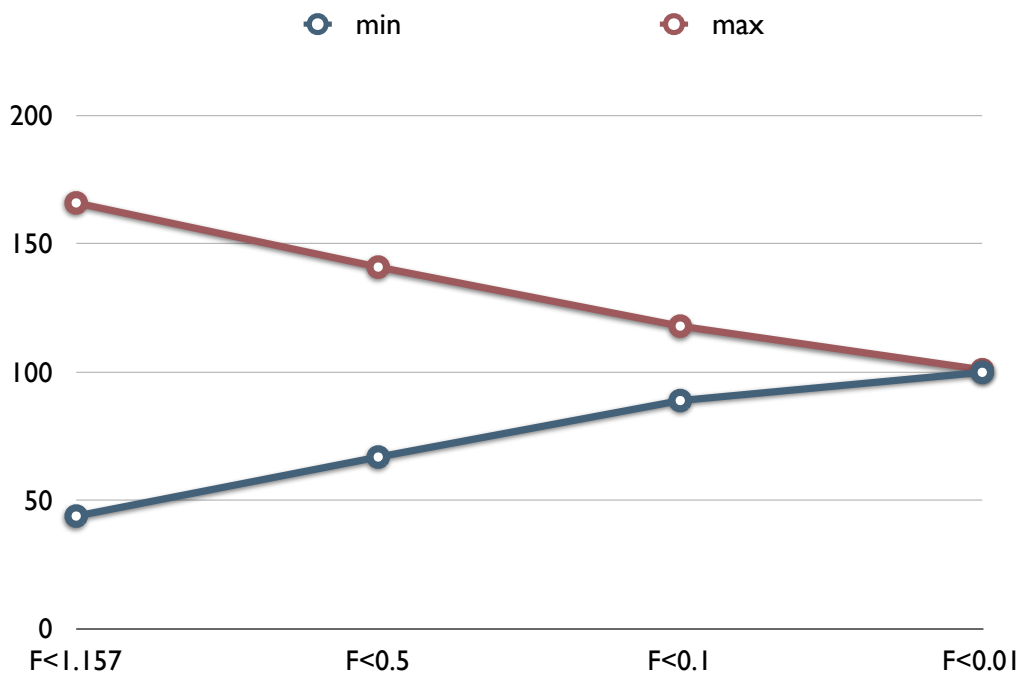


Figure 4: Visual expression of the parameter spread. Plotted are minimal and maximal values of 'optimal' parameters for \mathbf{x}_4 as example. Vertically between the two lines there are 'optimal' parameters in the sense that they give a cost function value F below the thresholds $\Delta F = 0.01, 0.1, 0.5, 1.165$. Minimal and maximal values are taken from tables 10, 11. Values of \mathbf{x}_4 on the vertical axis are scaled by 10^4 .

			$\Delta F = 0.1$				$\Delta F = 0.01$			
i	\mathbf{x}_i	\mathbf{x}_{di}	$\min(\mathbf{x}_i)$	$\max(\mathbf{x}_i)$	%	outlier $F = 11.5$	$\min(\mathbf{x}_i)$	$\max(\mathbf{x}_i)$	%	outlier $F = 0.15$
1	β	0.750	0.7029	0.8214	9.5	0.7029	0.7498	0.7600	1.3	0.76
2	μ_m	0.600	0.5499	0.6658	11.0	0.5499	0.5989	0.6100	1.7	0.598
3	α	0.025	0.0243	0.0291	16.6	0.0291	0.0250	0.0255	2.1	0.025
4	Φ_m^z	0.010	0.0089	0.0118	18.2	0.0118	0.0100	0.0101	0.8	0.010
5	κ	0.030	0.0284	0.0367	22.2	0.0284	0.0300	0.0312	3.9	0.0312
6	ϵ	1.000	0.9679	1.0560	5.6	0.9679	0.9994	1.0150	1.5	1.0150
7	g	2.000	1.7130	2.0000	14.4	1.7130	1.8100	2.0000	9.5	2.0000
8	Φ_m^p	0.010	0.0096	0.0112	12.2	0.0112	0.0100	0.0101	0.8	0.0101
9	Φ_z^*	0.205	0.1928	0.2252	9.9	0.2252	0.2050	0.2086	1.8	0.2050
10	γ_m	0.020	0.0195	0.0222	11.0	0.0195	0.0200	0.0201	0.6	0.0201
11	k_N	0.500	0.4033	0.6175	23.5	0.6175	0.4999	0.5344	6.9	0.5344
12	w_s	6.000	5.9950	6.5000	8.3	6.5000	6.0000	6.0080	0.1	6.0000

Table 11: As table 10, but with different values of ΔF .

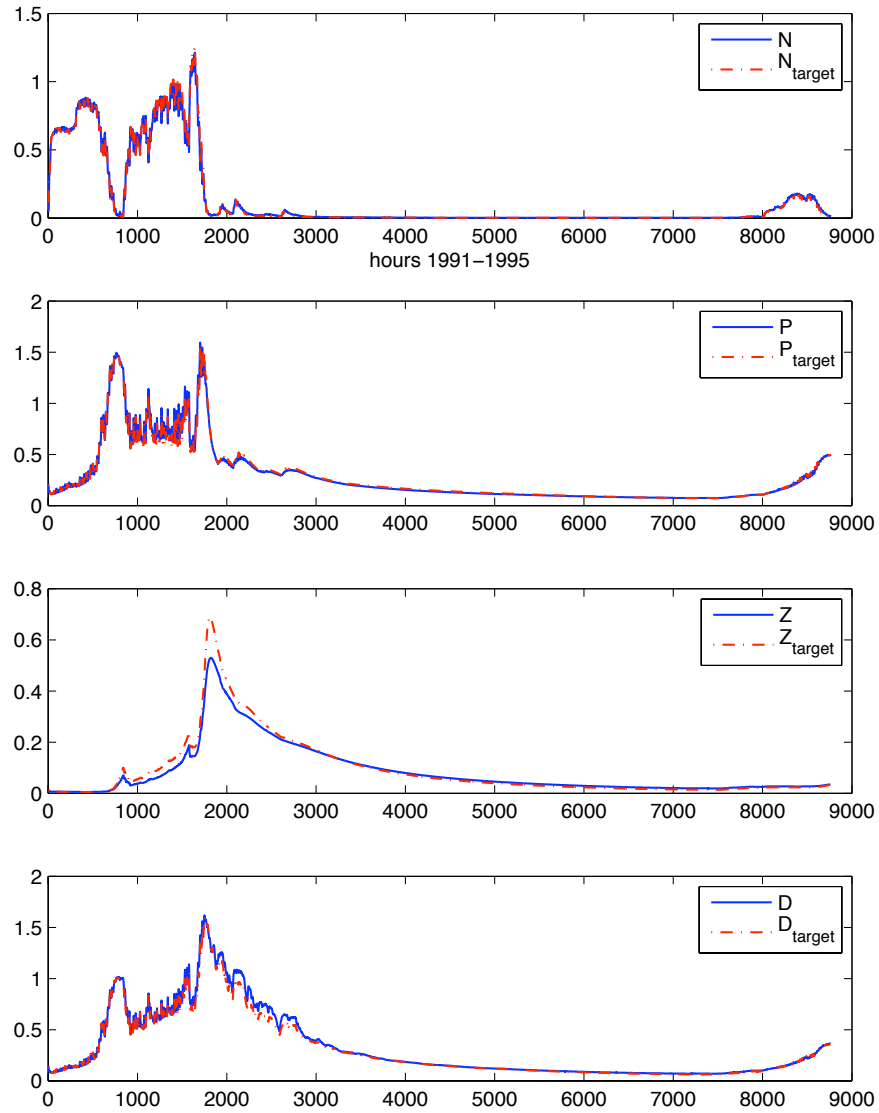


Figure 5: Trajectory of target parameter vector and hand-picked parameter vector with $F = 1.04$

			$\sigma_{pp} = 0.025$				$\sigma_{pp} = 0.3$			
i	\mathbf{x}_i	\mathbf{l}_i	\mathbf{u}_i	\mathbf{x}_i^*	\mathbf{u}_i	\mathbf{x}_i^*	\mathbf{u}_i	\mathbf{x}_i^*	\mathbf{u}_i	\mathbf{x}_i^*
1	β	0	1	1.000	1	1.000	1	1.000	1	1.000
2	μ_m	0.01	1.46	1.200	2.5	2.500	2.5	0.946	2.5	1.076
3	α	0	0.253	0.105	0.5	0.500	0.5	0.500	0.5	0.107
4	Φ_m^z	0	0.63	0.051	0.63	0.170	0.63	0.034	0.63	0.025
5	κ	0	0.073	0.073	0.2	0.172	0.2	0.160	0.2	0.026
6	ϵ	0.01	4	4.000	20	18.60	5	5.000	20	5.446
7	g	0.01	4	4.000	20	20.00	5	5.000	20	20.00
8	Φ_m^p	0.001	0.63	0.004	0.63	0.001	0.63	0.001	0.63	0.001
9	Φ_z^*	0	1	0.040	1	0.049	1	0.083	1	0.202
10	γ_m	0	0.15	0.150	0.15	0.000	0.15	0.150	0.15	0.092
11	k_N	0.01	1	0.100	2	1.760	2	2.000	2	1.827
12	w_s	0.1	128	105.0	128	6.200	128	120.3	128	23.01
	F			69.4		55.2		37.3		36.8

Table 12: Optimal parameters \mathbf{x}_i^* for lower bounds \mathbf{l}_i and different upper bounds \mathbf{u}_i and two different cost function weights σ_{PP} , see text. Changed bounds and changed optimal parameters are bold. Cost function time period was '91-'95 with spin-up year 1990. Initial profile for N was interpolation (Matlab[®] function interp1) of observational data 'around' 1.15.1991.

5.5 Best fit for observational data

We now turn to the realistic observational data obtained from the BATS source, see section 2. We applied again our optimization. For the parameter bounds as in [6], the same cost function values were obtained. Because some parameters are at the bounds, we also released the latter and ended up with optimization results as in table 12. Here we give some parameter vectors which gave the best fit for the observational data, depending on different parameter bounds prescribed. It can be seen that enlarging the upper bounds improves the fit, i.e. the cost function value. Notice that by enlarging the upper bounds, almost all parameters significantly change, not only those that were at the bounds. Using the weights σ given in section 2.5, we could not improve on $F = 55.2$ which does not mean that this is impossible. But we want to emphasize that increasing the width of the parameter bounds can and does allow for very unrealistic parameter estimates. This is regarded as a sign of model deficiency. Since [6] and [2] suggest a $\sigma_{pp} = 0.3$ for PP , we also give two parameter vectors for this cost function, also in table 12.

6 Conclusions

We presented a hybrid quantum-evolutionary and deterministic optimization method and applied it to a time-dependent, spatially one-dimensional marine biogeochemical model of *NPZD* type. The algorithm is parallelized and efficient in terms of computational effort. It turned out that both

- (1) a pure quantum-evolutionary version that only uses a very simple coordinate-oriented line search
- (2) and a nearly purely deterministic variant of the method, namely a gradient-based local SQP algorithm starting from randomly chosen parameter vectors,

prove to be suitable tools for parameter optimization with this model. Depending on the (directional) differentiability of the model, there is thus the choice to use only function evaluations in the evolutionary setting, or to apply techniques of Algorithmic or Automatic differentiation in the local, gradient-based method. The hybrid algorithm is flexible in the sense that it can easily be applied to other model of general type.

For this model, the local, gradient-based strategy with stochastically chosen starting values (version (2) above and as presented in section 3.3) works more efficient w.r.t. to computation time. We observed fast convergence to local minima and the best of all found minima is reached from approximately every third random starting point. Also the evolutionary-dominated version (see (1) above) succeeded in finding the minima.

Regarding the model under investigation, we summarize our results:

- If the given data to be fit are attainable (and thus assumed to be exact), the corresponding parameter vectors in many cases are unique, i.e. the discrete model is well suited for parameter identification.
- Taking into account data errors or uncertainties and thus accepting also parameters that yield cost function values above a given threshold, the in this sense also 'optimal' parameter vectors may lie in a wide range. This was already shown in [6]. Thus our results back up the propositions made therein, even more as we obtained them by a partially different and more general optimization strategy. The parameter spread seems *not* to be caused by too sparse data sampling.
- When optimizing the parameters to fit real observational data, the algorithm in both variants leads to non-optimal values. These values

depend on the chosen parameter bounds, which then have to be extended to values beyond reasonable biogeochemical modeling to obtain a better fit. Despite the massive computational power used and the corresponding exploration capabilities of the algorithm (when abstaining from fast local convergence), we were not able to find a better optimum. Thus we tentatively accept the found minima and argue that the *NPZD* model in the current form is very unlikely to give a better fit. A modification or extension of the model seems to be necessary.

- We also showed that the model output and thus an optimization depends on the chosen initial profiles and on the chosen spin-up period and length. We suggest to use interpolated observational data as initial profile and the corresponding date as starting point for the simulation/optimization. These mentioned errors are still dwarfed by the magnitude of the achieved cost F . To further validate the given model, the (estimated) physical forcings should be scrutinized.

We regard our method as flexible, reliable, and a valuable tool for parameter optimization. Moreover we think that the investigation of initial profile and spin-up sensitivities is a crucial step of model validation.

A Appendix: Modeling of Growth of Phytoplankton

The source minus sink equations of the *NPZD* model are effected by the light limited growth rate $\mu(z, t)$ of phytoplankton which varies with depth z and time t . Average light limited phytoplankton growth rates $\overline{\mu(k, t)}$ are calculated for every layer k of depth using a simplified version of an approximative formula by Evans and Parslow [4].

The basic formula for the light limited growth rate as a function of depth and time uses the curve of Smith, as recommended by Jassby and Platt when analytic integration is desired. The curve of Smith is applied to the variable irradiance:

$$\mu(z, t) = \frac{V_P \alpha I(z, t)}{\sqrt{V_P^2 + (\alpha I(z, t))^2}}$$

where V_P is the maximum growth rate of phytoplankton, α is the initial slope of photosynthesis vs light intensity, and $I(z, t)$ is the solar irradiance at depth z and time t . Evans and Parslow deal with a triangular approximation of the daily course of the sun leading to a double integral for the representation of average phytoplankton growth rate within one layer of depth.

In our case, time dependent values of the solar irradiance $I(0, t)$ at the ocean surface are taken from the physical model, and the solar incidence angle β_{air} at noon is assumed to be the equivalent daily averaged incidence angle for direct and diffuse radiation. The cosine of the solar incidence angle β_{water} in water corresponds to the relative way of light per depth and is calculated after Snells law [$\sin(\beta_{water}) = \sin(\beta_{air}/1.33)$]:

$$\cos(\beta_{water}) = \sqrt{1 - \frac{1 - \cos^2(\beta_{air})}{1.33^2}}.$$

The light attenuation factor per depth is supposed to be caused by water and phytoplankton only. For the k -th grid box, it is calculated as

$$\kappa(k, t) = \frac{1}{\cos(\beta_{water}) \cdot k_w} + \frac{\kappa \cdot P(k, t)}{\cos(\beta_{water})}.$$

With z_k and $z_k + 1$ as the top of the k -th and $(k + 1)$ -th box, respectively, we obtain

$$\overline{\mu(k, t)} = \frac{V_P}{z_{k+1} - z_k} \int_{z=z_k}^{z_{k+1}} \frac{\alpha I(0, t) e^{-\kappa(k, t)z}}{\sqrt{V_P^2 + (\alpha I(0, t) e^{-\kappa(k, t)z})^2}} dz.$$

Substitution of depth z by light intensity $\varphi(z) = \alpha I(0, t) e^{-\kappa(k, t)z}$ gives

$$\overline{\mu(k, t)} = -\frac{V_P}{\kappa(k, t)(z_{k+1} - z_k)} \int_{y=\varphi(z_k)}^{\varphi(z_{k+1})} \frac{1}{\sqrt{V_P^2 + y^2}} dz$$

with analytical solution

$$\overline{\mu(k, t)} = \ln \frac{\varphi(z_k) + \sqrt{V_P^2 + \varphi(z_k)^2}}{\varphi(z_{k+1}) + \sqrt{V_P^2 + \varphi(z_{k+1})^2}}.$$

Acknowledgements

The authors would like to thank Andreas Oschlies and Iris Kries, IfM Geomar, Kiel. This research was supported by DFG Cluster of Excellence Future Ocean, Project A3 and Miniproposal: Mathematical and Algorithmic Challenges in Modelling Biochemical Cycles.

References

- [1] Schartau, M., Oschlies, A., 2003. *Simultaneous data-based optimization of a 1D-ecosystem model at three locations in the North Atlantic: Part I - Method and parameter estimates*. Journal of Marine Research 61, 765-793.

- [2] Oschlies, A., Garcon, V., 1999. *An eddy-permitting coupled physical-biological model of the North Atlantic. 1. Sensitivity to advection numerics and mixed layer physics.* Global Biogeochemical Cycles 13, 135-160.
- [3] Oschlies, A., Koeve, W., Garcon, V. 2000. *An eddy-permitting coupled physical-biological model of the North Atlantic. 2. Ecosystem dynamics and comparison with satellite and JGOFS local studies data.* Global Biogeochemical Cycles 14, 499-523.
- [4] Evans, G. T., Parslow, J. S., 1985. *A model of annual plankton cycles.* Biological Oceanography 3, 328-347.
- [5] A. Griewank: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, No. 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.
- [6] Ward, Ben A., 2009. *Marine Ecosystem Model Analysis Using Data Assimilation.* PhD Thesis, <http://web.mit.edu/benw/www/Thesis.pdf>.
- [7] Roman, M. R., Caron, D. A., Kremer, P., Lessard, E. J., Madin, L. P., Malone, T. C., Napp, J. M., Peele, E. R., Youngbluth, M. J., 1995. *Spatial and temporal changes in the partitioning of organic carbon in the plankton community of the Sargasso Sea off Bermuda.* Deep-Sea Research I 42 (6), 973-992.
- [8] Schartau, Markus *Data-assimilation studies of marine, nitrogen based, ecosystem models in the North Atlantic Ocean.* PhD Thesis.
- [9] Babu, G. S. S., Das, D. B., Patvardhan, C., 2008. *Real-parameter Quantum Evolutionary Algorithm for Economic Load Dispatch.* IET Proceedings on Generation, Transmission and Distribution.
- [10] Han, K. H., Kim, J. H., 2002. *Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization.* IEEE Transactions on Evolutionary Computation, 6 (6), 580-593.
- [11] www.fastopt.com
- [12] www.aemdesign.com