

Technische Universität Berlin
Institut für Mathematik

**Parameter optimization using algorithmic
differentiation in a reduced-form model of
the Atlantic thermohaline circulation**

Thomas Slawig, Kirsten Zickfeld^a

^aPotsdam-Institut f. Klimafolgenforschung, Postfach 601203, 14412 Potsdam

Preprint 2003/02

Parameter optimization using algorithmic differentiation in a reduced-form model of the Atlantic thermohaline circulation

Thomas Slawig, Kirsten Zickfeld*

January 9, 2003

Abstract

Optimizing some model parameters a reduced-form model of the Atlantic thermohaline circulation (THC) is fitted to data provided by a comprehensive climate model. Different techniques to compute stationary states of the reduced-form model are discussed. The fitting problem is formulated as weighted least squares optimization problem with non-linear constraints that enforce a proper representation of the present climate. Possible formulations of the optimization problem are presented and compared with respect to their numerical treatment. The technique of Algorithmic or Automatic Differentiation (AD) is used to provide gradient information that can be used in the optimization. The application of the AD software is described in detail and numerical results are given.

Key words: Parameter identification, Ocean modeling, Algorithmic differentiation

1 Introduction

The Atlantic thermohaline circulation (THC) is the part of the Atlantic Ocean circulation which is driven by density gradients, as opposed to the wind-driven component. The northern branches of the THC, the Gulf Stream and the North Atlantic Drift, are commonly known to be responsible for the relatively mild climate of the North Atlantic region. Paleo-reconstructions [1] and physical considerations [15] suggest that the THC is non-linear, with a well defined threshold beyond which the circulation switches off. Model simulations have shown that anthropogenic climate change could act as a trigger for such a mode switch [5, 14, 11]. Because of the possibly severe consequences that a collapse of the THC could bring about, analysis of the political and socio-economic implications associated with such a scenario is currently high on the research agenda [4]. Such analysis is often conducted by means of so called ‘integrated assessment models’ (IAMs), i.e., models of the coupled climate-economy system [6, 17]. Since state-of-the-art climate models representing the THC are often too complex to be incorporated into IAMs, there is the need to develop computationally efficient model components which, although highly simplified compared to the comprehensive models, are able to reproduce key features of their time-dependent (or ‘transient’) behavior [20, 12].

In this paper we describe the parameter optimization of a ‘box model’ of the THC. The model describes the dynamics of the THC by a system of eight non-linear ordinary differential equations for temperature and salinity in four boxes. The box model parameters are optimized with respect to data obtained with CLIMBER-2, a climate model of intermediate complexity which has proven to successfully describe key elements of past and present climates, including the THC [8, 2].

The optimization is performed in a strictly mathematical sense, i.e., as a least squares minimization problem with non-linear equality and inequality constraints. This problem is solved by a modern Sequential Quadratic Programming (SQP) method available in the software package

*Potsdam-Institut f. Klimafolgenforschung, Postfach 601203, 14412 Potsdam

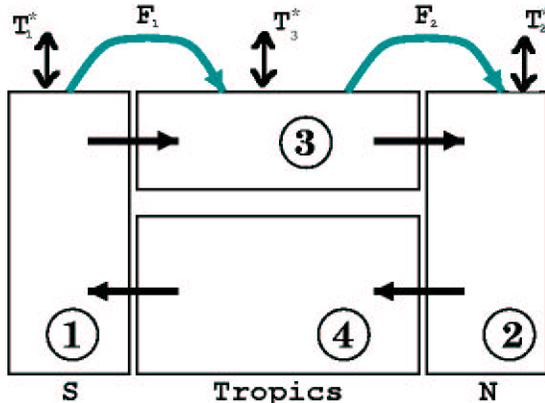


Figure 1: Schematic of the four-box model of the Atlantic thermohaline circulation. The temperatures of boxes 1, 2, and 3 are relaxed toward the values T_1^* , T_2^* , and T_3^* , respectively. The salinities are forced by the fresh water fluxes F_1 and F_2 . The meridional flow is proportional to the density gradient between boxes 1 and 2.

MATLAB¹. This optimization method makes use of available gradient information of the cost functional and the constraints. To achieve high accuracy and optimal performance we supply the exact derivatives of both cost functional and constraints by using a library for algorithmic differentiation (AD).

We only use available numerical solution and optimization software from the MATLAB package which was applied efficiently and successfully. The improvement or the development of new software is not the aim of this work.

The paper is structured as follows: We start with a short description of the reduced-form model of the THC. We then present the aim of the parameter optimization. We derive two variants of the steady state equations of the model and enter a discussion of possible solution methods. In the following section we discuss different formulations of the parameter optimization problem. We then describe the basic ideas and the used software library for algorithmic computation of exact derivatives of the cost functional and the constraints and its application to the reduced-form model. Finally we compare the algorithmically computed gradients with finite difference approximations and present the numerical solution of the parameter optimization problem, i.e. the optimal fit. We compare the different variants of the steady state equations with respect to their usability in the optimization.

2 The reduced-form model

The model is an inter-hemispheric box model of the Atlantic which has been successfully applied for the investigation of bifurcations and the stability of the Atlantic thermohaline circulation [10, 13, 16]. The model configuration is shown in Fig. 1. It consists of four well-mixed basins, representing the southern, tropical, northern, and deep Atlantic, respectively. Neighboring boxes are connected to allow for a continuous, closed-loop circulation. The surface boxes are linked to the overlying atmosphere through fluxes of heat and fresh water. Assuming the water in the northern box is denser than the water in the southern box, a pressure-driven circulation develops with northward flow near the surface and southward flow at depth. Once the circulation is active, a positive feedback mechanism becomes operational. Indeed, the transport of salt from the Tropics towards the northern latitudes increases the salinity and hence the density of the North Atlantic waters. The latter, in turn, leads to a stronger north-south density gradient and an intensified

¹MATLAB is a registered trademark of The Mathworks Inc. Natick, MA, US.

circulation. This picture is consistent with the current understanding of the ‘real world’ THC and lays out the reason for its non-linearity.

In this four-box model the meridional volume transport m (or ‘overturning’) is proportional to the density difference $\rho_2 - \rho_1$ between boxes 1 and 2:

$$m = k(\rho_2 - \rho_1) = k[\beta(S_2 - S_1) - \alpha(T_2 - T_1)]. \quad (1)$$

Here $S_2 - S_1$ and $T_2 - T_1$ are the north-south salinity and temperature gradients, k is a hydraulic constant linking volume transport m to the density difference, α and β are thermal and haline expansion coefficients, respectively.

The temperatures and salinities of the four boxes adjust to the oceanic transport of heat and fresh water. Further, temperatures and salinities of the surface boxes are forced by the overlying atmosphere through fluxes of heat and fresh water. The surface heat fluxes are described by a restoring law of the form $Q \sim (T^* - T)$, while salinity forcing consists of fixed atmospheric vapor transports F_i between the upper boxes. This leads to the following set of ordinary differential equations for temperatures T_i and salinities S_i of each of the four boxes:

$$\dot{T}_1 = \frac{m}{V_1}(T_4 - T_1) + \lambda_1(T_1^* - T_1) \quad (2)$$

$$\dot{T}_2 = \frac{m}{V_2}(T_3 - T_2) + \lambda_2(T_2^* - T_2) \quad (3)$$

$$\dot{T}_3 = \frac{m}{V_3}(T_1 - T_3) + \lambda_3(T_3^* - T_3) \quad (4)$$

$$\dot{T}_4 = \frac{m}{V_4}(T_2 - T_4) \quad (5)$$

$$\dot{S}_1 = \frac{m}{V_1}(S_4 - S_1) + S_0 \frac{F_1}{V_1} \quad (6)$$

$$\dot{S}_2 = \frac{m}{V_2}(S_3 - S_2) - S_0 \frac{F_2}{V_2} \quad (7)$$

$$\dot{S}_3 = \frac{m}{V_3}(S_1 - S_3) + S_0 \frac{F_2 - F_1}{V_3} \quad (8)$$

$$\dot{S}_4 = \frac{m}{V_4}(S_2 - S_4) \quad (9)$$

Here V_i are box volumina and T_i^* the temperatures the surface boxes are restored to. F_1 and F_2 are the fresh water fluxes (multiplied by a reference salinity, S_0 , for conversion to a salt flux) into the tropical and northern Atlantic, respectively. The λ_i are thermal coupling constants and are derived as functions of the thermal coupling Γ and the box thickness z_i :

$$\lambda_i = \frac{\Gamma}{c\rho_0 z_i},$$

where c is the specific heat capacity of sea water and ρ_0 its density.

3 Aim of the optimization

As already pointed out in the introduction, the goal of our work is to tune the box model such that it is able to reproduce key features of the behavior of comprehensive models. We have chosen CLIMBER-2 as reference model since it is computational efficient and allows to perform a multitude of runs in a reasonable time frame. Our intention is to tune the box model parameters such that not only the present-day state of the Atlantic ocean as simulated by CLIMBER-2 is reproduced, but also key features of its dynamic behavior as, for example, the location of the threshold beyond which the THC shuts down (the ‘bifurcation point’). We therefore determine the unknown model parameters (i.e., the restoring temperatures T_1^* , T_2^* and T_3^* , the flow constant k , and the thermal coupling constant Γ) by fitting the box model to a ‘hysteresis’ experiment with CLIMBER-2.

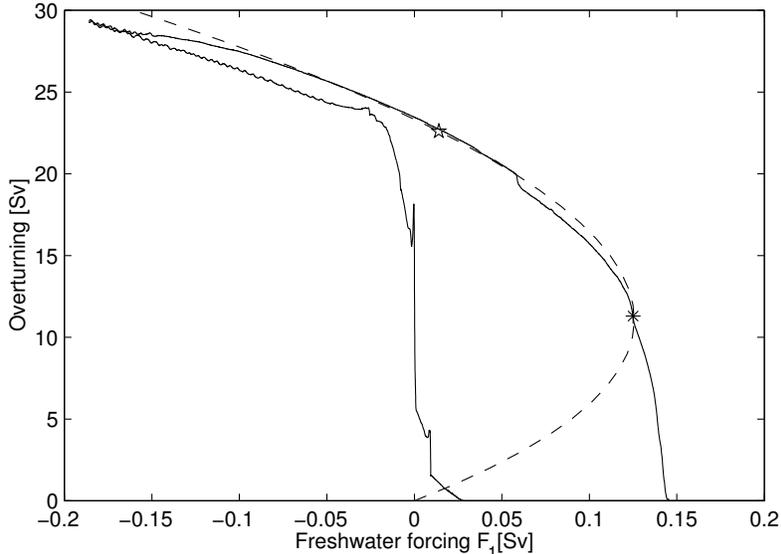


Figure 2: Steady state solutions $m(F_1)$ of the box model (with optimized parameters, dashed curve) and hysteresis curve derived with CLIMBER-2 (solid curve). The star and the asterisk denote the present-day state and the bifurcation point, respectively.

Such a hysteresis experiment is performed by slowly increasing and successively decreasing the fresh water flux F_1 into the North Atlantic (cf. [9]). The idea is that the rate at which the fresh water perturbation is applied (i.e., $0.05 \cdot 10^{-3}$ Sv/yr; $1 \text{ Sv} = 10^6 \text{ m}^3/\text{s}$) is so small that the system is always in a state close to equilibrium. The hysteresis traces the response of the volume transport to these changes in the fresh water flux. The curve obtained with the CLIMBER-2 model is depicted in Figure 2.

The upper branch of the hysteresis is computed by *increasing* the fresh water flux F_1 into the North Atlantic. The curve shows that the higher F_1 , the faster the decrease in the volume transport. As soon as F_1 reaches a value of about $F_1 = 0.14$, the circulation collapses, indicating that the bifurcation point has been hit. Once the circulation is zero, the fresh water flux is *decreased*. As becomes evident in the figure, the circulation does not recover until the fresh water perturbation reaches zero. Thus the CLIMBER-2 model predicts a hysteresis behavior of the thermohaline volume transport with respect to the fresh water flux F_1 .

Since we are mainly interested in reproducing the present-day overturning state and the bifurcation point we take only the upper branch of the hysteresis curve as data set $(F_{1j}, \bar{m}_j)_{j=1, \dots, N}$. Here the number of available data pairs is $N = 8000$. Since the last values of the volume transport data \bar{m}_j are very close to zero, they are omitted, giving a number of now $N = 6634$ with $\bar{m}_j \geq 10^{-6}$. To reduce the computational effort we start with a smaller, equidistantly distributed set of data, taking only one pair out of 100.

Because the data represent *quasi-equilibrium* values of the CLIMBER-2 model, the parameters of the box model are optimized in such a way that the *stationary values* of the volume transport of the box model are close to the data. Therefore in the next section we derive expressions for the steady state solutions of the box model.

4 The steady state of the reduced-form model

The steady state equations of the box model can be derived from equations (1) to (9) by setting the time derivatives to zero. We briefly describe how the stationary equation for the volume transport

m is derived without giving all the algebraic computations:

- From equations (5) and (9) one easily obtains $T_4 = T_2$ and $S_4 = S_2$.
- Using the latter and (6) in (1), the volume transport m can be expressed as function just of the two variables T_1, T_2 and the model parameters as a solution to

$$m^2 + k\alpha(T_2 - T_1)m + k\beta S_0 F_1 = 0.$$

- Using (3) and (4) T_3 and T_1 can be expressed only via m and the difference $\Delta T := T_2 - T_1$. Then (2) can be exploited to express the difference ΔT just using m and the parameters.
- Equations (6) to (8) can be used to compute the differences $S_2 - S_1, S_3 - s_1$ etc. from m . Note that (8) as the sum of (6) and (7) is redundant in the stationary case.

Note that the stationary values of the temperatures T_i are unique, whereas the salinities S_i are only determined up to the same additive constant.

Using $\lambda_i V_i = \Gamma \frac{V_i}{c\rho_0 z_i}$ and the abbreviations

$$\mu_i := \frac{V_i}{c\rho_0 z_i}, \quad \mu_{ij} := \mu_i \mu_j, \quad \mu := \mu_{12} + \mu_{13} + \mu_{23}, \quad \sigma := \sum_{i=1}^3 \mu_i, \quad \tau := \prod_{i=1}^3 \mu_i$$

we get the following expression for the temperature difference $\Delta T = T_2 - T_1$:

$$\Delta T = \Gamma \frac{m(T_2^* \mu_{12} + T_3^* \mu_{13} - T_1^* (\mu_{12} + \mu_{13})) + (T_2^* - T_1^*) \Gamma \tau}{m^2 \sigma + m \Gamma \mu + \Gamma^2 \tau}. \quad (10)$$

Hence for given value of the fresh water flux F_1 the stationary values of the volume transport m are given as the solutions to the rational equation

$$G_1(m) := m^2 + k\alpha \Gamma \frac{a_1 m + a_0}{\sigma m^2 + \Gamma \mu m + \Gamma^2 \tau} m + k\beta S_0 F_1 = 0 \quad (11)$$

with

$$a_1 = T_2^* \mu_{12} + T_3^* \mu_{13} - T_1^* (\mu_{12} + \mu_{13}), \quad a_0 = (T_2^* - T_1^*) \Gamma \tau.$$

Critical points in (11) are the zeros of the denominator of the second term, i.e. poles of the function. We can show the following:

Proposition 4.1 *If $\mu_i \geq 0$ for $i = 1, 2, 3$ then for all $m \geq 0$ and all F_1 function G_1 has no poles.*

Proof: Note that $\mu, \sigma, \tau, \Gamma$ are non-negative. The roots of the denominator in the second term of (11) are

$$m_{1,2} = -\frac{\Gamma \mu}{2\sigma} \pm \sqrt{\frac{\Gamma^2 \mu^2}{4\sigma^2} - \frac{\Gamma^2 \tau}{\sigma}} = \frac{\Gamma}{2\sigma} \left(-\mu \pm \sqrt{\mu^2 - 4\tau\sigma} \right).$$

For $\mu^2 < 4\tau\sigma$ these roots are both complex. In the case $\mu^2 \geq 4\tau\sigma$ we have $\sqrt{\mu^2 - 4\tau\sigma} \leq \sqrt{\mu^2} = \mu$ and thus both roots are negative. \square

For the current setting of the μ_i the roots are complex numbers. Since the model only makes sense for non-negative values of the volume transport m the proposition above implies that (11) can be equivalently written as

$$G_2(m) := c_4 m^4 + c_3 m^3 + c_2 m^2 + c_1 m + c_0 = 0 \quad (12)$$

where now

$$\begin{aligned}
c_4 &= \sigma, \\
c_3 &= \Gamma\mu, \\
c_2 &= \Gamma^2\tau + k\alpha\Gamma[T_2^*\mu_{12} + T_3^*\mu_{13} - T_1^*(\mu_{12} + \mu_{13})] + k\beta S_0 F_1 \sigma \\
c_1 &= k\Gamma[\alpha\Gamma(T_2^* - T_1^*)\tau + \beta S_0 F_1 \mu] \\
c_0 &= k\Gamma^2\beta S_0 F_1 \tau.
\end{aligned}$$

Because of $\Gamma, \mu_i > 0$ for $i = 1, 2, 3$ we get

$$c_4, c_3 > 0 \quad \text{and} \quad \text{sign}(c_0) = \text{sign}(F_1). \quad (13)$$

The sign of the coefficients c_1, c_2 remains undetermined even if the sign of F_1 is known; it depends on the relations between the T_i^* .

Computation of steady states

To compute the steady value of m for given freshwater flux F_1 (with the other parameters fixed) equation (11) or (12) has to be solved. Vice versa equation (11) can be used to determine F_1 as a function of m :

$$F_1 = F(m) := -\frac{1}{k\beta S_0} \left(m^2 + k\alpha\Gamma \frac{a_1 m + a_0}{\sigma m^2 + \Gamma\mu m + \Gamma^2\tau} m \right). \quad (14)$$

Since this equation is explicit it is clearly more convenient than treating (11) or (12). We keep this in mind for the optimization but first discuss the numerical differences between (11) and (12): At a first glimpse equation the latter seems to be preferable to (11) since a fourth order polynomial compared to a rational function is somehow "simpler". Its roots can be computed even explicitly by *Ferrari's method* [3]. Moreover numerical solution techniques via a corresponding eigenvalue problem or by applying iterative solvers directly on G_2 are available (the MATLAB routines `root` and `fzero`, respectively). Since Ferrari's method may include complex arithmetic (even if a root is actually real) the numerical methods are preferable from a practical point of view.

Anyhow computations show that the coefficients (compare Table 1) are such that both G_1 and G_2 have very small values between the two positive zeros, compare Table 1 and 2, and Figures 3 and 4.

μ	$\sigma = c_4$	τ	$\sigma m^2 + \Gamma\mu m + \Gamma^2\tau$	c_3	c_2	c_1	c_0	a_1	a_0
2e-5	9e-2	2e-8	4e-6	2e-4	-3e-5	-2e-6	5e-11	-1e-4	6e-3

Table 1: Magnitude of the coefficients of G_1, G_2 (for $F_1 = 0.014$) and F for the parameter values $(T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$. The differences in the magnitude remain the same for all relevant values of the fresh water flux $F_1 \in [-0.2, 0.15]$. Values of m are scaled since the time-dependent box model computes in years model time whereas F_1 and m are given in $Sv = 10^6 m^3/s$, i.e. per second.

$G_1(m=0)$	$\min_{[0,30]} G_1(m)$	$G_1(m=30)$	$G_2(m=0)$	$\min_{[0,30]} G_2(m)$	$G_2(m=30)$
3.1e-06	-4.8e-05	2.8e-05	4.6e-11	-1.3e-09	1.3e-09

Table 2: Magnitude of G_1 and G_2 for $m \in [0, 30]$ and the same parameter values as in Table 1.

One of the two positive zeros lies near zero, the other one is closer to the data \bar{m}_j and thus is the one that is appropriate for the optimization. But due to the scales of G_1, G_2 in the range

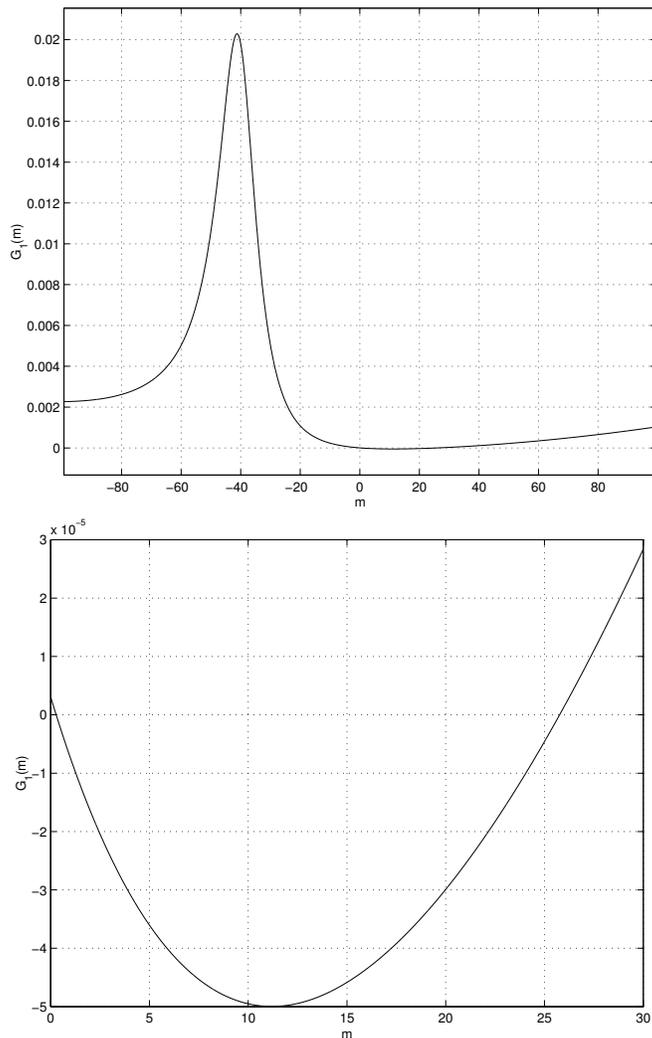


Figure 3: Top: Graph of G_1 as function of m for $F_1 = 0.014$ and the parameter values $(T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$, bottom: zoomed in the relevant range $m \in [0, 30]$.

$m \in [0, 30]$ the sensitivity of the zero with respect to changes in the parameters is very high. Moreover the chosen equation (11) or (12) has to be solved with a high accuracy to get a good result for the corresponding stationary value of m . Thus G_1 is preferable to G_2 because the latter is even six orders of magnitude lower. Hence numerical solution schemes to approximate a root of G_2 have to be run with a very small stopping criterion. For some numerical methods (as e.g. MATLAB's routine `root`) this is not possible. As a result approximations obtained via the corresponding eigenvalue problem lead to unusable, very inaccurate approximations of the zero. It is thus necessary to compute the zeros directly from (11) or (12). Because of Proposition 4.1 the advantage of the polynomial structure of (12) then is not relevant anymore and G_1 can be taken as well. In the numerical examples we will present and discuss results for both choices.

5 Mathematical formulation of the optimization problem

In this section we present the exact mathematical formulation of the parameter optimization problem. We denote by \vec{x} the vector of parameters to be optimized, in our case $\vec{x} = (T_1^*, T_2^*, T_3^*, k, \Gamma)$.

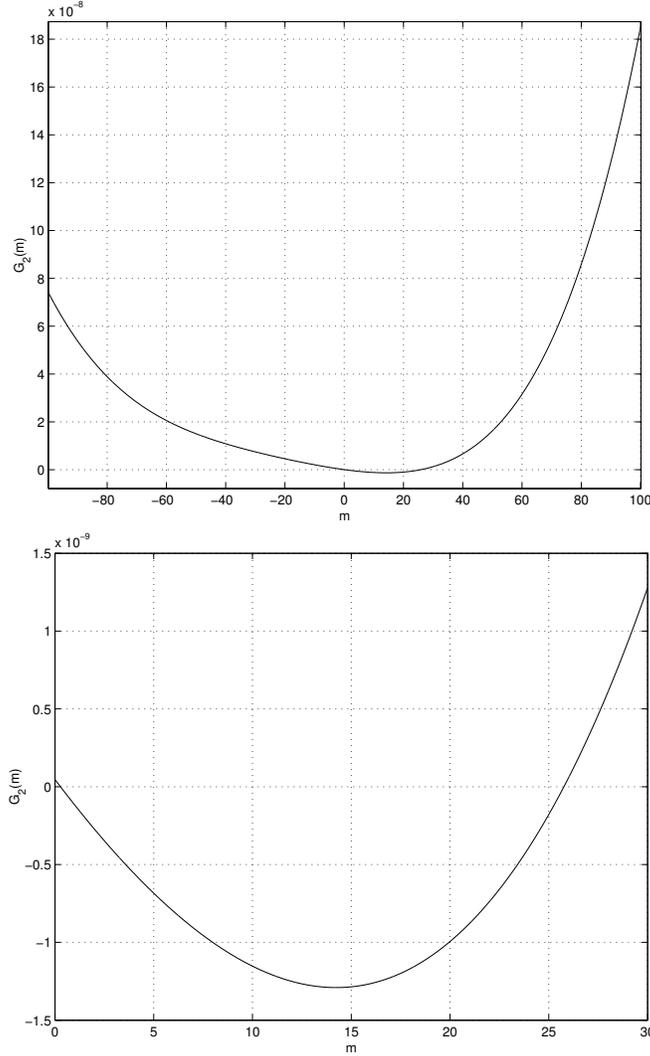


Figure 4: Graph of G_2 for the same parameters as in Figure 3.

Recalling that a steady state of the box model can be expressed

- either by computing m as a function of F_1 via (11) or (12)
- or vice versa writing F_1 as function of m using (14)

we may write the optimization problem in different formulations. We discuss them here with a special emphasis on the numerical treatment by an iterative, gradient-based optimization algorithm as for example MATLAB's SQP method.

- Regarding the steady state m as a function of F_1 we obtain the problem

$$\min_{\vec{x}} J_1(\vec{x}) := \frac{1}{2} \sum_{j=1}^N \omega_j (m_j - \bar{m}_j)^2 \quad (15)$$

where $m_j := m(F_{1j}, \vec{x})$ solves $G_k(F_{1j}, m_j, \vec{x}) = 0, j = 1, \dots, N$

for $k = 1$ or 2 . Here each evaluation of J_1 requires N solutions of the state equation $G_k = 0$.

Computing the gradient we obtain by the chain rule

$$\nabla_x J_1(\vec{x}) = \sum_{j=1}^N \omega_j (m(F_{1j}, \vec{x}) - \bar{m}_j) \nabla_x m(F_{1j}, \vec{x}).$$

The crucial point here is to compute the gradient of m with respect to \vec{x} . A numerical differentiation of the iterative solver to compute a root of G_1, G_2 is not recommended. Instead of that one of the implicit equations (11) or (12) is differentiated with respect to \vec{x} . Omitting the argument F_{1j} this gives

$$\nabla_x G_k(m(\vec{x}), \vec{x}) + \frac{\partial G_k}{\partial m}(m(\vec{x}), \vec{x}) \nabla_x m(\vec{x}) = 0.$$

Thus we obtain for the gradient of m with respect to \vec{x} :

$$\nabla_x m(\vec{x}) = - \left(\frac{\partial G_k}{\partial m}(m(x), \vec{x}) \right)^{-1} \nabla_x G_k(m(\vec{x}), \vec{x}). \quad (16)$$

Both partial derivatives on the right can be evaluated without differentiating the iterative solver of the equation $G_k = 0$.

- An alternative approach which is standard in control and optimization is to take both $\vec{m} := (m_j)_{j=1, \dots, N}$ and \vec{x} as control variables, define

$$J_2(\vec{m}, \vec{x}) := \frac{1}{2} \sum_{i=j}^N \omega_j (m_j - \bar{m}_j)^2$$

and incorporate the coupling between \vec{x} and \vec{m} as additional constraint:

$$\min_{(\vec{m}, \vec{x})} J_2(\vec{m}, \vec{x}) \quad \text{s.t.} \quad G_k(F_{1j}, m_j, \vec{x}) = 0, \quad j = 1, \dots, N. \quad (17)$$

Even though J_2 does not depend on x directly in this case, we keep the notation $J_2(\vec{m}, \vec{x})$. This approach is more flexible in the sense that an iterative numerical optimization algorithm may chose values of (\vec{m}, \vec{x}) during the iteration that satisfy the state equation $G_k = 0$ not exactly, but only up to a certain accuracy. This *constraint violation* may lead to better results and/or performance, a fact that we observe in our numerical computations. The maximal constraint violation in an optimization algorithm usually can be prescribed. Some optimization routines (e.g. the one from MATLAB's Optimization Toolbox we use, see below) only allow the user to prescribe one value for all constraints. To be more flexible here we replace the equality constraints $G_k(F_{1j}, m_j, \vec{x}) = 0$ by

$$G_k(F_{1j}, m_j, \vec{x})^2 \leq \delta_j, \quad j = 1, \dots, N \quad (18)$$

such that the δ_j can be used to adjust the desired accuracy point-wise. The gradient of the cost functional is given by

$$\frac{\partial J_2}{\partial m_j}(\vec{m}, \vec{x}) = \omega_j (m_j - \bar{m}_j), \quad j = 1, \dots, N, \quad \nabla_x J_2(\vec{m}, \vec{x}) = 0.$$

Moreover the gradient of the constraint (18) has to be computed, i.e.

$$\frac{\partial G_k}{\partial m}(F_{1j}, m_j, \vec{x}) \quad \text{and} \quad \nabla_x G_k(F_{1j}, m_j, \vec{x}).$$

Here only partial derivatives of G_k are needed. Due to the de-coupling of \vec{m} and \vec{x} in the problem formulation (17) the dependency of m_j on \vec{x} has not to be taken into account. This is one important advantage of formulation (17) compared to treating (15) directly.

- Regarding F_1 as a function of m and the parameter vector \vec{x} in the steady model, i.e. equation (14), we get

$$\min_{\vec{x}} J_3(\vec{x}) := \frac{1}{2} \sum_{j=1}^N \omega_j (F(\bar{m}_j, \vec{x}) - F_{1j})^2. \quad (19)$$

Since F is an explicit function of m we compute the gradient as

$$\nabla_x J_3(\vec{x}) = \sum_{j=1}^N \omega_j (F(\bar{m}_j, \vec{x}) - F_{1j}) \nabla_x F(\bar{m}_j, \vec{x}).$$

Up to now the last formulation seems to be superior to the others since the problem is unconstrained, the function F is explicit, and its gradient with respect to \vec{x} can be computed easily.

Additional constraints

The stationary values $T_{ie}, i = 1, 2, 3$, of the temperatures shall fulfill the following constraints at the state of the present climate ($F_1 = F_e := 0.014$):

$$T_{1e} := \frac{m_e \Delta T_e}{\Gamma \mu_1} + T_1^* \in [5, 7] \quad (20)$$

$$T_{3e} := \frac{\Gamma \mu_3 T_3^* + m_e T_{1e}}{\Gamma \mu_3 + m_e} = 11.4 \quad (21)$$

$$T_{2e} := \frac{\Gamma \mu_2 T_2^* + m_e T_{3e}}{\Gamma \mu_2 + m_e} = 4.7 \quad (22)$$

where $\Delta T_e := T_{2e} - T_{1e}$ is given by (10) and $m_e := m(F_e)$. Moreover physical restrictions $T_1^*, T_3^*, k, \Gamma > 0$ have to be satisfied. T_2^* may be negative.

The two equality constraints may be used to eliminate two of the parameters analytically. But since the resulting form of (20) for the remaining parameters is not simpler than the original constraints this gives no advantage in the numerical optimization. Even more the option of (small) constraint violations may give better results in an optimization. Thus we use the constraints in the original form (20) – (22). Using non-linear functions c_j^{eq}, c_j they are written as

$$c_j^{eq}(m_e, \vec{x}) = 0, \quad c_j(m_e, \vec{x}) \leq 0, \quad j = 1, 2.$$

Since $m_e = m_e(F_e)$ now either equation (11) or (12) has to be used. For the computation of the gradients of the constraints with respect to \vec{x} the same situation arises as for problem formulations (15) or (17): Either one of the implicit equations for G_1 or G_2 has to be differentiated using (16), or \bar{m} and \vec{x} are de-coupled and only the partial derivatives with respect to m and \vec{x} of the constraints have to be computed.

This implies that due to the constraints the gain of using the explicit function F in (14) loses its advantage. For this reason and because of the higher flexibility already mentioned above formulation (17) is used for the optimization.

Optimization algorithm

The problem is solved by the routine `fmincon` from MATLAB's optimization toolbox [7]. It uses a SQP (Sequential Quadratic Programming) method that incorporates the constraints by Lagrange multipliers. The method approximates the cost functional by a sequence of quadratic sub-problems. Since the weighted least squares functional is quadratic itself this is somehow redundant. MATLAB also provides a linear and non-linear least squares solvers, but both do not treat non-linear constraints. Thus the SQP method is used here. A possible overhead is not crucial here since the problem size is still rather small and the computation time remains reasonable, at

least if not all data points (F_{1j}, \bar{m}_j) are taken. The MATLAB routine makes use of provided derivative information of cost functional and constraints with respect to the optimization variables, i.e. (\bar{m}, \bar{x}) . The partial derivatives above are computed exactly, without any approximation error, by the technique of Algorithmic Differentiation (AD) that we describe in the following section.

6 Algorithmic Differentiation

Algorithmic (or Automatic) Differentiation (AD) is a software technology to compute the *exact* derivative of a function given in the form of a computer programme, i.e. without any approximation errors. This is an important advantage compared for example to finite differences (FD). In this section we describe the basic concepts of AD, explain the technique of the used AD software, and show how it is applied to the box model code.

The code of the box (or any other) model can be seen as a function

$$y = F(x), \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^m,$$

mapping a vector of independent input variables $x \in \mathbb{R}^n$ to a vector of dependent output variables $y \in \mathbb{R}^m$. The function is a (long) concatenation of elementary intrinsic functions and operators F_i (e.g. $+$, $-$, \sin etc.) of the used programming language. The computation of the output y is described by

$$x_0 := x, \quad x_i := F_i(x_{i-1}), i = 1, \dots, k, \quad y := x_k \quad (23)$$

where the x_i are intermediate variables lying on the path from input x to output y . Each F_i can be differentiated *exactly* by standard rules of calculus. By the chain rule the derivative of F then can be written as the matrix product

$$y' := F'(x) = F'_k(x_{k-1}) \cdots F'_2(x_1) F'_1(x)$$

or recursively as

$$x'_0 := I, \quad x'_i := F'_i(x_{i-1})x'_{i-1}, i = 1, \dots, k, \quad y' := x'_k \quad (24)$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. Combining value and derivative in a new variable $X_i := (X_i^{(1)}, X_i^{(2)}) := (x_i, x'_i)$ we may compute (23) and (24) simultaneously as

$$X_0 := (x, I), \quad (25)$$

$$X_i := \left(F_i(X_{i-1}^{(1)}), F'_i(X_{i-1}^{(1)})X_{i-1}^{(2)} \right), \quad i = 1, \dots, k, \quad (26)$$

$$y = X_k^{(1)}, \quad y' = X_k^{(2)}. \quad (27)$$

The evaluation of $F_i(x_{i-1})$ in (23) in each step is replaced by the application of a new function defined by (26) which operates on X_i . This can be interpreted and realized as an *overloading* of F_i for the new datatype to which X_i belongs.

The software library ADMAT ([18], [19]) provides this functionality for MATLAB. It contains overloaded versions of MATLAB operators and functions. The datatype for the X_i is called `deriv`. Then only (25) has to be inserted as

```
x = deriv(x, eye(size(x)));
```

before the (unchanged) model computations. Here `x` is the vector of inputs and `eye` denotes the identity matrix in MATLAB. At the end (27) is realized by calls to the intrinsic ADMAT functions `getval()` and `getydot()`.

As a typical example for the computation of a partial derivative we take the j -th constraint in (18) which is implemented in MATLAB as

```

g=G(F1(j), [m(j), x]);
c(j)=g^2-delta(j);

```

Since we want to compute the derivatives with respect to (m, x) simultaneously the functions G_k (G in the code) are written in such a way that they take $[m(j), x]$ as one input vector. Using the chain rule we compute only the inner derivative of G_k algorithmically. We thus replace the above lines by

```

g=G(F1(j), deriv([m(j), x], eye(1+length(x))));
c(j)=getval(g)^2-omega(j);
Gc([j, n+1:n+length(x)], j)=2*getval(g)*getydot(g)';

```

Here $Gc(:, j)$ denotes the gradient of the constraint $c(j)$ as it is required by the optimization routine `fmincon` in MATLAB. The usage of the AD library ADMAT for the other constraints is similar.

7 Numerical results

In this section we compare the algorithmically generated derivatives with those approximated by finite differences (FD). Here we concentrate on the derivatives of G_k with respect to m and \vec{x} . Then we present numerical results obtained for the parameter optimization problem.

The accuracy of the AD-generated derivatives compared to FD derivatives is sufficiently high, see Table 3. The FD computations show that the choice of the step-size is very delicate. We observe the smallest differences between both derivatives for a rather big step-size of 10^{-2} , whereas for smaller values (down to 10^{-8}) the differences were bigger. This is due to the magnitudes of G_1 ($\approx 10^{-5}$) and G_2 ($\approx 10^{-10}$). Thus even for relatively big step-sizes the range of machine precision is reached. MATLAB's optimization methods also provide an option with a gradient check to test user-provided derivatives (as e.g. generated by AD) against those computed by FD. In all runs this test is successful as well. Thus the AD derivatives here prove to be a robust tool for optimization. Moreover they are efficient since they save about 20 % cputime.

	FD step-size	difference	$\frac{\partial G_k}{\partial m}$	$\frac{\partial G_k}{\partial T_1^*}$	$\frac{\partial G_k}{\partial T_2^*}$	$\frac{\partial G_k}{\partial T_3^*}$	$\frac{\partial G_k}{\partial k}$	$\frac{\partial G_k}{\partial \Gamma}$
G_1	1e-2	absolute	1e-17	5e-23	7e-23	5e-23	6e-23	4e-16
	1e-2	relative	2e-06	5e-14	2e-13	2e-11	6e-13	7e-07
	1e-8	absolute	9e-17	1e-16	1e-16	4e-17	5e-17	5e-17
	1e-8	relative	2e-06	9e-08	1e-07	1e-05	8e-07	2e-07
G_2	1e-2	absolute	2e-12	3e-18	4e-18	6e-18	1e-18	1e-12
	1e-2	relative	1e-06	1e-13	3e-13	6e-11	7e-13	9e-07
	1e-8	absolute	4e-12	5e-12	7e-12	4e-12	1e-12	6e-12
	1e-8	relative	8e-07	4e-07	4e-07	4e-06	1e-06	5e-06

Table 3: Absolute and relative (w.r.t. AD derivative) differences between AD-generated and central finite difference gradient of G_1 and G_2 . Shown are the maximum vector norms over 67 equidistantly distributed data points (F_{1j}, \bar{m}_j) for the parameter values $\vec{x} = (T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$.

Using the formulation (17) of the optimization problem the size of the problem becomes much bigger since now $\vec{m} \in \mathbb{R}^N$ is a variable, too. Therefore it is recommended to start with a subset of the data points, and increase the number of data points successively. We use every 100th point of the original data. Moreover we exclude those where the volume transport is close to zero ($\bar{m}_j < 10^{-6}$). It turns out that also the part of the upper branch of the hysteresis curve where the slope becomes too big (that is for $F_1 > 0.125$) have to be excluded to obtain a good fit. We

started using $N = 63$ data points. Later on we incorporate more and more data points, but this does not give any improvement.

Using G_1 to represent the steady values of m the optimization takes about 560 iterations to proceed to a very satisfying solution. As can be seen in Table 4 and Fig. 5 the large number of iterations is necessary to improve the fit near the point of collapsing. Further iterations result in infeasible solutions since a stationary value of m close to the data is not found. This again indicates the sensitivity of the problem, specifically in finding the appropriate root of G_1 . When using G_2 this problem occurred after 530 iterations, but the last feasible solution is much worse, compare again Table 4. Using G_1 this point is already reached after 200 iterations, and then the algorithm continues successfully. Thus G_1 should be chosen to characterize the steady states of the model.

	it.	T_1^*	T_2^*	T_3^*	k	Γ	\hat{J}	m_e	T_{1e}	T_{2e}	T_{3e}
G_1	0	6.40	-2.00	12.00	25.00	10.00	0.0	22.72	5.80	3.45	11.25
	100	6.25	0.79	12.01	35.61	11.93	32.1	22.60	5.98	4.70	11.40
	200	6.27	1.25	11.94	34.25	13.51	27.9	22.60	6.02	4.70	11.40
	300	6.41	2.01	11.80	30.11	17.34	20.4	22.60	6.19	4.70	11.40
	400	6.46	2.19	11.77	28.87	18.58	18.5	22.60	6.25	4.70	11.40
	500	6.54	2.45	11.73	27.22	20.71	16.1	22.60	6.34	4.70	11.40
	562	6.64	2.68	11.69	25.43	23.06	13.9	22.60	6.45	4.70	11.40
G_2	530	6.24	1.16	11.95	35.17	13.16	25.5	22.60	6.00	4.70	11.40
bound		≥ 0		≥ 0	≥ 0	[10, 75]		[22.6, 22.8]	[5, 7]	4.70	11.40

Table 4: Convergence behavior for problem (17) using formulation (18) for the constraint.

As can be seen from Fig. 5 the fits obtained during the optimization are very good in the middle range where $F_1 \in [-0.1, 0.06]$. The left part of the hysteresis curve is not fitted well, a fact that can not be improved by adjusting the weights ω_j and/or δ_j . But this area is not very interesting for further transient computations: It corresponds to a climate state with strong fresh water export from the Atlantic which is not expected to be reached in the wake of global warming. The present climate is fitted very well, which is natural since the additional constraints enforce a good match here. The kink in the hysteresis curve at $F_1 \approx 0.06$ can not be fitted by the reduced-form model. Moreover there remains a discrepancy between the data and the fitted curve in the range $F_1 \geq 0.04$. The optimization improves the behavior of the stationary box model near the point of collapsing at the right end of the curve. Since the fitted curve at $F_1 = 0.014 = F_e$ is with $m_e = 22.6$ already at the lower bound for this constraint, a further improvement is difficult. But even relaxing this lower bound for m_e further and adjusting the weights ω_j in this area do not lead to better results. Thus summarizing the weights ω_j, δ_j in this examples were a useful method to test the limits of the model.

8 Summary

In this paper we perform a parameter optimization for a reduced-form model of the Atlantic thermohaline circulation. The output of the stationary model is fitted to data provided by quasi-stationary runs with CLIMBER-2, a climate model of intermediate complexity. Additional constraints concerning the present climate state are imposed. We derive alternative equations characterizing the stationary state, show their equivalence, and discuss their differences with respect to numerical solution techniques. Here we use and compare available software in MATLAB. We discuss different formulations of the optimization problem and show how to solve it using an SQP method from MATLAB's Optimization Toolbox. For the computation of descent directions we use exact derivative information obtained by using a library for Algorithmic or Automatic Differentiation (AD). The AD-generated derivatives show very good agreement with finite difference approximations. Moreover they are successfully used in the optimization and lead to about 20%

saving in cputime. The results of the optimization show a significant improvement in the fit of the reduced-form model compared to the initial parameter values, specifically in the most interesting area, namely the bifurcation point and the present-day state of the Atlantic thermohaline circulation.

Acknowledgments

The authors would like to thank Stefan Rahmstorf, developer of the box model, Christian Meyer and Stefan Kaphengst who performed some of the computations, and Arun Verma for his kind support in using ADMAT. The work of the second author was supported by the German Federal Ministry of Science, grant no. 01LD0016.

References

- [1] W. Dansgaard, S.J. Johnsen, H.B. Clausen, N.S. Dahl-Jensen, N.S. Gundestrup, C.U. Hammer, C.S. Hvidberg, J.P. Steffensen, A.E. Sveinbjornsdottir, J. Jouzel and G. Bond: Evidence for general instability of past climate from a 250-kyr ice-core record, *Nature* 364, 1993, 218-220.
- [2] A. Ganopolski, S. Rahmstorf, V. Petoukhov and M. Claussen: Simulation of modern and glacial climates with a coupled global model of intermediate complexity, *Nature* 391, 1998, 351-356.
- [3] D. Herbison-Evans: Caterpillars and the Inaccurate Solution of Cubic and Quartic Equations, *Australian Computer Science Communications*, Vol. 5, No. 1(1983), see also [<http://www-staff.mcs.uts.edu.au/~don/pubs/solving.html>].
- [4] INTEGRATION: Integrated assessment of changes in the thermohaline circulation, [<http://www.pik-potsdam.de/~stefan/Projects/integration/index.html>].
- [5] S. Manabe and R.J. Stouffer: Century-scale effects of increased atmospheric CO₂ on the ocean-atmosphere system, *Nature* 364, 1993, 215-218.
- [6] W.D. Nordhaus: *Managing the global commons: the economics of climate change*, MIT Press, Cambridge, MA, 1994.
- [7] *Optimization Toolbox For Use with MATLAB User's Guide Version 2*, The Mathworks Inc., Natick, MA, USA 2000.
- [8] V. Petoukhov, A. Ganopolski, V. Brovkin, M. Claussen, A. Eliseev, C. Kubatzki and S. Rahmstorf: CLIMBER-2: a climate system model of intermediate complexity. Part I: model description and performance for present climate, *Climate Dynamics* 16, 2000, 1-17.
- [9] S. Rahmstorf: Bifurcations of the Atlantic thermohaline circulation in response to changes in the hydrological cycle, *Nature* 378, 1995, 145-149.
- [10] S. Rahmstorf: On the freshwater forcing and transport of the Atlantic thermohaline circulation, *Climate Dynamics* 12, 1996, 799-811.
- [11] S. Rahmstorf and A. Ganopolski: Long-term global warming scenarios computed with an efficient coupled climate model, *Climatic Change* 43, 1999, 353-367.
- [12] S.H. Schneider and S.L. Thompson: Simple climate model used in economic studies of global change, in: S.J. DeCanio and R.B. Howarth and A.H. Sanstad and S.H. Schneider and S.L. Thompson (eds.), *New directions in the economics and integrated assessment of global climate change*, Pew Center on Global Climate Change, Arlington, VA, 2000, 59-80.

- [13] J.R. Scott, J. Marotzke and P.H. Stone: Interhemispheric thermohaline circulation in a coupled box model, *J. of Physical Oceanography* 29, 1999, 351-365.
- [14] T.F. Stocker and A. Schmittner: Influence of CO₂ emission rates on the stability of the thermohaline circulation, *Nature* 388, 1997, 862-865.
- [15] H. Stommel: Thermohaline convection with two stable regimes of flow, *Tellus* 13, 1961, 224-241.
- [16] S. Titz, T. Kuhlbrodt, U. Feudel and S. Rahmstorf: On freshwater-dependent bifurcations in box models of the interhemispheric thermohaline circulation, *Tellus* 54(1), 2002, 89-98.
- [17] F.L. Toth, T. Bruckner, H.-M. Füssel, M. Leimbach and G. Petschel-Held: Integrated assessment of long-term climate policies: Part 1 - Model presentation, *Climatic Change*, 2003, forthcoming.
- [18] A. Verma: ADMAT project homepage, Cornell Univ., Ithaca, NY, USA, [<http://www.tc.cornell.edu/~averma/AD/ADMAT.html>].
- [19] A. Verma: ADMAT: Automatic Differentiation in MATLAB using object oriented methods, [<http://www.tc.cornell.edu/~averma/AD/admatoo.ps>].
- [20] K. Zickfeld, S. Rahmstorf and T. Slawig: A reduced-form model for the response of the Atlantic thermohaline circulation to global warming, 2003, submitted.

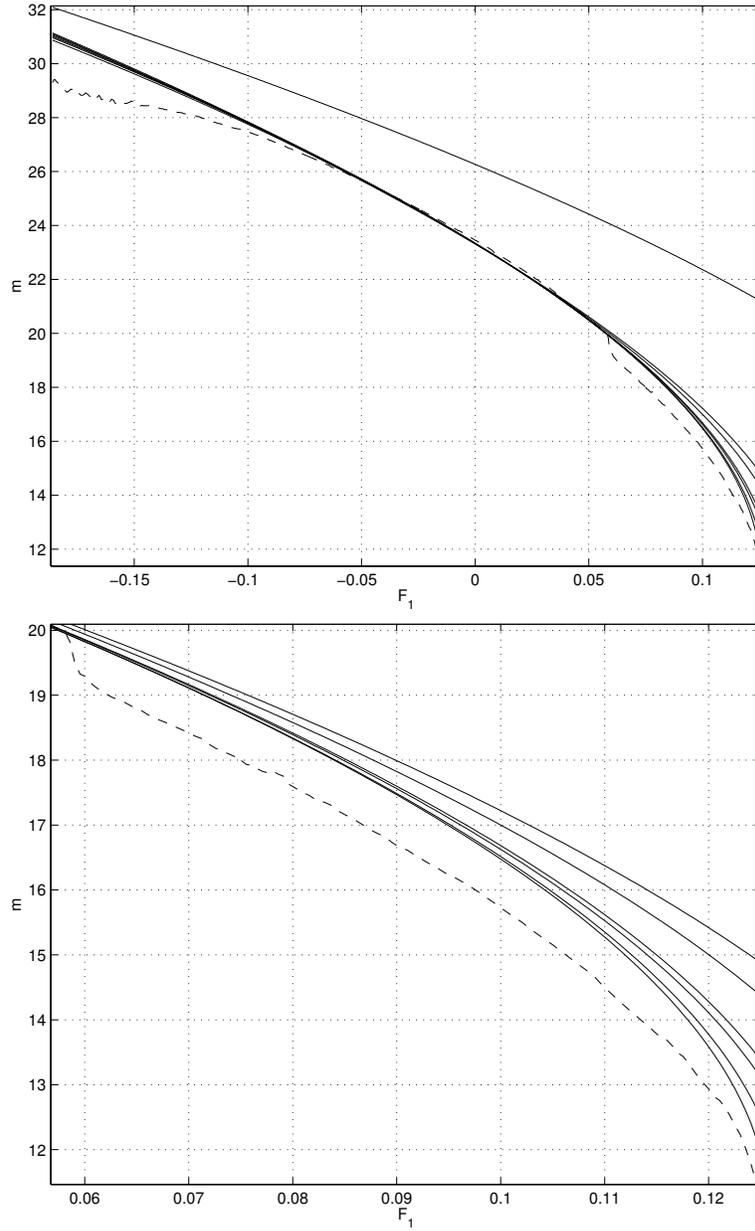


Figure 5: Top: Fitted curves of stationary volume transport $m(F_1)$ for the box model. Dashed: data, solid: curves obtained with optimized parameters at the start and after 100, 200, 300, 400, 500, 562 iterations (from top to bottom, compare Table 4), bottom: zoomed near the point of collapsing.