



ELSEVIER

Nonlinear Analysis: Real World Applications ■■■ (■■■■) ■■■–■■■

**Nonlinear  
Analysis**

Real World Applications

www.elsevier.com/locate/na

1  
3  
5  
7  
9

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

Thomas Slawig<sup>a,\*</sup>, Kirsten Zickfeld<sup>b</sup>

<sup>a</sup>*Institut für Mathematik, Technische Universität Berlin, MA-45 Straße des 17. Juni 136, Berlin 10623, Germany*

<sup>b</sup>*Potsdam Institute for Climate Impact Research, P.O. Box 601203, Potsdam 14412, Germany*

Received 20 November 2002; accepted 23 December 2003

---

## Abstract

11 Optimizing some model parameters a reduced-form model of the Atlantic thermohaline cir-  
13 culation is fitted to data provided by a comprehensive climate model. Different techniques to  
15 compute stationary states of the reduced-form model are discussed. The fitting problem is for-  
17 mulated as weighted least-squares optimization problem with non-linear constraints that enforce  
19 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.

© 2003 Published by Elsevier Ltd.

21 *Keywords:* Parameter identification; Ocean modeling; Algorithmic differentiation

---

## 1. Introduction

23 The Atlantic thermohaline circulation (THC) is the part of the Atlantic Ocean circula-  
25 tion 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  
27 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.

---

\* Corresponding author. Tel.: +49-30-31428036; fax: +49-30-31479706.

E-mail address: [slawig@math.tu-berlin.de](mailto:slawig@math.tu-berlin.de) (T. Slawig).

1 Model simulations have shown that anthropogenic climate change could act as a trigger  
2 for such a mode switch [5,11,14]. Because of the possibly severe consequences that a  
3 collapse of the THC could bring about, analysis of the political and socio-economic  
4 implications associated with such a scenario is currently high on the research agenda  
5 [4]. Such analysis is often conducted by means of the so-called integrated assessment  
6 models (IAMs), i.e., models of the coupled climate–economy system [6,17]. Since  
7 state-of-the-art climate models representing the THC are often too complex to be in-  
8 corporated into IAMs, there is the need to develop computationally efficient model  
9 components which, although highly simplified compared to the comprehensive models,  
10 are able to reproduce key features of their time-dependent (or ‘transient’) behavior  
11 [12,20].

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

18 The optimization is performed in a strictly mathematical sense, i.e., as a least-squares  
19 minimization problem with non-linear equality and inequality constraints. This problem  
20 is solved by a modern sequential quadratic programming (SQP) method available in the  
21 software package MATLAB.<sup>1</sup> This optimization method makes use of available gradi-  
22 ent information of the cost functional and the constraints. To achieve high accuracy  
23 and optimal performance we supply the exact derivatives of both cost functional and  
24 constraints by using a library for algorithmic differentiation (AD).

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

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

## 39 2. The reduced-form model

40 The model is an inter-hemispheric box model of the Atlantic which has been success-  
41 fully applied for the investigation of bifurcations and the stability of the Atlantic THC

---

<sup>1</sup> MATLAB is a registered trademark of The Mathworks Inc., Natick, MA, US.

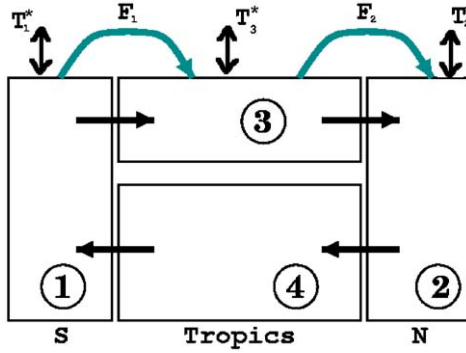


Fig. 1. Schematic of the four-box model of the Atlantic THC. The temperatures of boxes 1, 2, and 3 are relaxed toward the values  $T_1^*$ ,  $T_2^*$ , and  $T_3^*$ , respectively. The salinities are forced by the fresh water fluxes  $F_1$  and  $F_2$ . The meridional flow is proportional to the density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

1 [10,13,16]. The model configuration is shown in Fig. 1. It consists of four well-mixed  
 2 basins, representing the southern, tropical, northern, and deep Atlantic, respectively.  
 3 Neighboring boxes are connected to allow for a continuous, closed-loop circulation.  
 4 The surface boxes are linked to the overlying atmosphere through fluxes of heat and  
 5 fresh water. Assuming the water in the northern box is denser than the water in the  
 6 southern box, a pressure-driven circulation develops with northward flow near the sur-  
 7 face and southward flow at depth. Once the circulation is active, a positive feedback  
 8 mechanism becomes operational. Indeed, the transport of salt from the Tropics towards  
 9 the northern latitudes increases the salinity and hence the density of the North Atlantic  
 10 waters. The latter, in turn, leads to a stronger north-south density gradient and an in-  
 11 tensified circulation. This picture is consistent with the current understanding of the  
 12 ‘real world’ THC and lays out the reason for its non-linearity.

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

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

15 Here  $S_2 - S_1$  and  $T_2 - T_1$  are the north–south salinity and temperature gradients,  $k$  is  
 16 a hydraulic constant linking volume transport  $m$  to the density difference,  $\alpha$  and  $\beta$   
 17 are thermal and haline expansion coefficients, respectively.

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

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

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

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

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

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

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

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

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

- 1 Here  $V_i$  are box volumina and  $T_i^*$  the temperatures the surface boxes are restored to.  $F_1$   
 3 and  $F_2$  are the fresh water fluxes (multiplied by a reference salinity,  $S_0$ , for conversion  
 5 to a salt flux) into the tropical and northern Atlantic, respectively. The  $\lambda_i$  are thermal  
 coupling constants and are derived as functions of the thermal coupling  $\Gamma$  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  $\rho_0$  its density.

### 7 3. Aim of the optimization

As already pointed out in the Introduction, the goal of our work is to tune the box  
 9 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  
 11 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  
 13 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  
 15 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  
 17 the thermal coupling constant  $\Gamma$ ) by fitting the box model to a ‘hysteresis’ experiment  
 with CLIMBER-2.

19 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  
 21 rate at which the fresh water perturbation is applied (i.e.,  $0.05 \times 10^{-3}$  Sv/yr; 1 Sv =  
 $10^6$  m<sup>3</sup>/s) is so small that the system is always in a state close to equilibrium. The  
 23 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 Fig. 2.

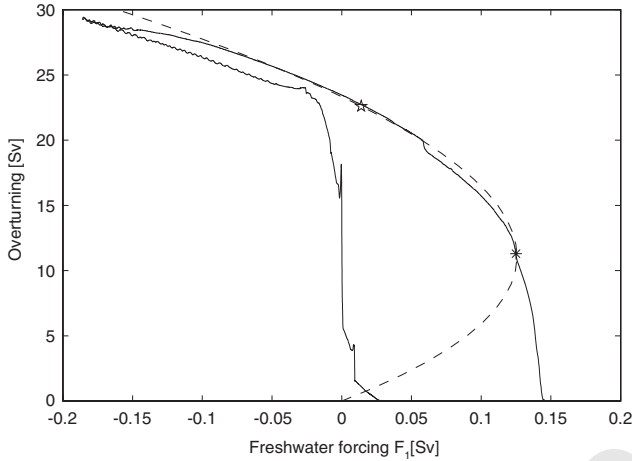


Fig. 2. Steady-state solutions  $m(F_1)$  of the box model (with optimized parameters, dashed curve) and hysteresis curve derived with CLIMBER-2 (solid curve). The star and the asterisk denote the present-day state and the bifurcation point, respectively.

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

9 Since we are mainly interested in reproducing the present-day overturning state and  
 11 the bifurcation point we take only the upper branch of the hysteresis curve as data set  
 13  $(F_{1j}, \bar{m}_j)_{j=1, \dots, N}$ . Here the number of available data pairs is  $N = 8000$ . Since the last  
 15 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.

17 Because the data represent *quasi-equilibrium* values of the CLIMBER-2 model, the  
 19 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

21 The steady-state equations of the box model can be derived from Eqs. (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 Eqs. (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  $\Delta T$  just using  $m$  and the parameters.
  - Eqs. (6)–(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  $\lambda_i V_i = \Gamma(V_i/c\rho_0 z_i)$  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  $\Delta T = T_2 - T_1$ :

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

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

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

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.  $\square$

Table 1

Magnitude of the coefficients of  $G_1, G_2$  (for  $F_1 = 0.014$ ) and  $F$  for the parameter values  $(T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$

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

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  $\text{Sv} = 10^6 \text{ m}^3/\text{s}$ .

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

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

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

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

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

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

9 *4.1. Computation of steady-states*

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

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

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

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

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

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

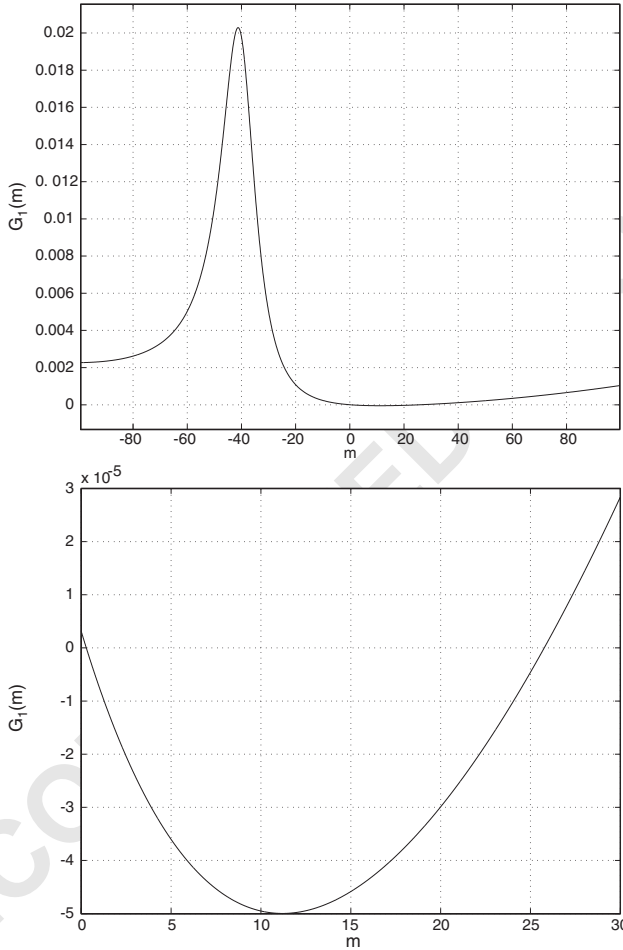


Fig. 3. Top: Graph of  $G_1$  as function of  $m$  for  $F_1 = 0.014$  and the parameter values  $(T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$ , bottom: zoomed in the relevant range  $m \in [0, 30]$ .

- 1 One of the two positive zeros lies near zero, the other one is closer to the data  $\bar{m}_j$
- 3 and thus is the one that is appropriate for the optimization. But due to the scales of  $G_1, G_2$  in the range  $m \in [0, 30]$  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

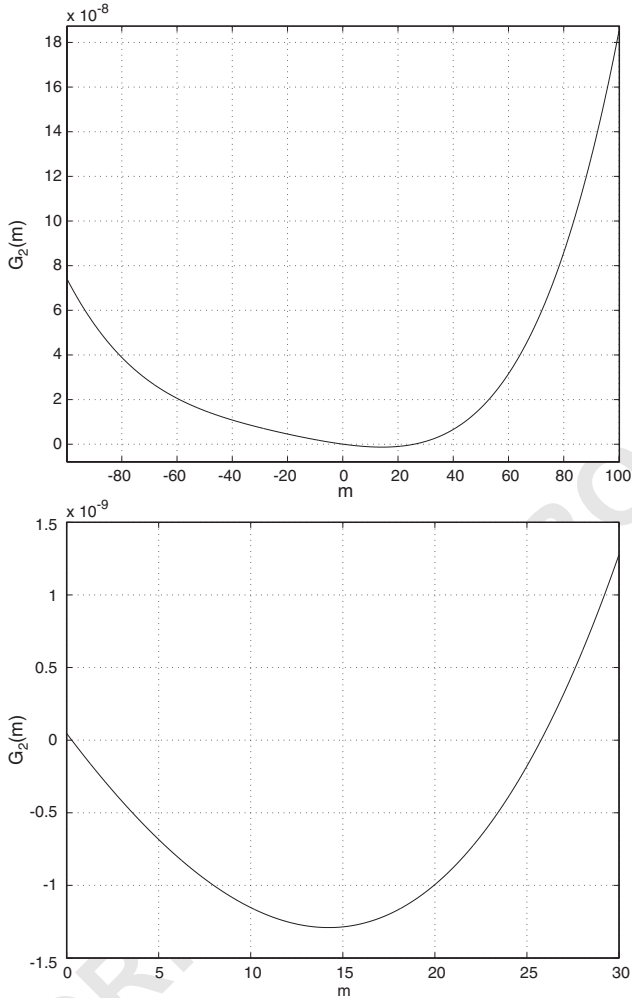


Fig. 4. Graph of  $G_2$  for the same parameters as in Fig. 3.

- 1 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.  
 3 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  
 5 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  
 7 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  
 9  $G_1$  can be taken as well. In the numerical examples we will present and discuss results  
 for both choices.

1 **5. Mathematical formulation of the optimization problem**

3 In this section we present the exact mathematical formulation of the parameter opti-  
 3 mization problem. We denote by  $\vec{x}$  the vector of parameters to be optimized, in our case  
 $\vec{x} = (T_1^*, T_2^*, T_3^*, k, \Gamma)$ . Recalling that a steady-state of the box model can be expressed

- 5 • 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)

7 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.

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

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

where  $m_j := m(F_{1j}, \vec{x})$  solves  $G_k(F_{1j}, m_j, \vec{x}) = 0, j = 1, \dots, N$  (15)

11 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

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

13 The crucial point here is to compute the gradient of  $m$  with respect to  $\vec{x}$ . A numerical  
 15 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

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

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

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

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

- 21 • 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 an additional constraint:

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

1 Even though  $J_2$  does not depend on  $x$  directly in this case, we keep the notation  
 2  $J_2(\vec{m}, \vec{x})$ . This approach is more flexible in the sense that an iterative numerical op-  
 3 timization algorithm may chose values of  $(\vec{m}, \vec{x})$  during the iteration that satisfy the  
 4 state equation  $G_k = 0$  not exactly, but only up to a certain accuracy. This *constraint*  
 5 *violation* may lead to better results and/or performance, a fact that we observe in  
 6 our numerical computations. The maximal constraint violation in an optimization  
 7 algorithm usually can be prescribed. Some optimization routines (e.g. the one from  
 8 MATLAB's Optimization Toolbox we use, see below) only allow the user to pre-  
 9 scribe 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

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

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

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

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

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

15 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  $\vec{x}$  is not to be taken into  
 17 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,  
 19 i.e. Eq. (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

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

21 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  
 23 computed easily.

### 5.1. Additional constraints

25 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 Eq. (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  $\vec{m}$  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.

### 5.2. Optimization algorithm

The problem is solved by the routine `fmincon` from MATLAB's optimization toolbox [7]. It uses a SQP 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  $(F_{1j}, \vec{m}_j)$  are taken. The MATLAB routine makes use of provided derivative information of cost functional and constraints with respect to the optimization variables, i.e.  $(\vec{m}, \vec{x})$ . The partial derivatives above are computed exactly, without any approximation error, by the technique of 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

1 to finite differences (FD). In this section we describe the basic concepts of AD, explain  
 2 the technique of the used AD software, and show how it is applied to the box model  
 3 code.

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

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

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

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

9 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  
 11 the derivative of  $F$  then can be written as the matrix product

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

or recursively as

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

13 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

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

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

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

15 The evaluation of  $F_i(x_{i-1})$  in (23) in each step is replaced by the application of a new  
 16 function defined by (26) which operates on  $X_i$ . This can be interpreted and realized as  
 17 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 con-  
 19 tains overloaded versions of MATLAB operators and functions. The datatype for the  $X_i$   
 is called `deriv`. Then only (25) has to be inserted as

$$21 \quad \mathbf{x} = \text{deriv}(\mathbf{x}, \text{eye}(\text{size}(\mathbf{x}))),$$

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

25 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

$$27 \quad \mathbf{g} = \mathbf{G}(\mathbf{F1}(\mathbf{j}), [\mathbf{m}(\mathbf{j}), \mathbf{x}]),$$

$$\mathbf{c}(\mathbf{j}) = \mathbf{g}^{\wedge} 2 - \text{delta}(\mathbf{j}),$$

Table 3

Absolute and relative (w.r.t. AD derivative) differences between AD-generated and central FD gradient of  $G_1$  and  $G_2$

	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

Shown are the maximum vector norms over 67 equidistantly distributed data points  $(F_{1j}, \bar{m}_j)$  for the parameter values  $\vec{x} = (T_1^*, T_2^*, T_3^*, k, \Gamma) = (6.4, -2, 12, 25, 10)$ .

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

```

5   g=G(F1(j),deriv([m(j),x],eye(1+length(x))))),
   c(j)=getval(g)^ 2-omega(j),
7   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  
 9 optimization routine `fmincon` in `MATLAB`. The usage of the AD library `ADMAT` for the  
 other constraints is similar.

## 11 7. Numerical results

12 In this section we compare the algorithmically generated derivatives with those ap-  
 13 proximated by FD. Here we concentrate on the derivatives of  $G_k$  with respect to  $m$  and  
 14  $\vec{x}$ . Then we present numerical results obtained for the parameter optimization problem.

15 The accuracy of the AD-generated derivatives compared to FD derivatives is suffi-  
 16 ciently high, see Table 3. The FD computations show that the choice of the step-size  
 17 is very delicate. We observe the smallest differences between both derivatives for a  
 18 rather big step-size of  $10^{-2}$ , whereas for smaller values (down to  $10^{-8}$ ) the differences  
 19 were bigger. This is due to the magnitudes of  $G_1$  ( $\approx 10^{-5}$ ) and  $G_2$  ( $\approx 10^{-10}$ ). Thus  
 20 even for relatively big step-sizes the range of machine precision is reached. `MATLAB`'s  
 21 optimization methods also provide an option with a gradient check to test user-provided

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

	It.	$T_1^*$	$T_2^*$	$T_3^*$	$k$	$\Gamma$	$\hat{J}$	$m_e$	$T_{1e}$	$T_{2e}$	$T_{3e}$
$G_1$	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		$\geq 0$		$\geq 0$	$\geq 0$	[10, 75]		[22.6, 22.8]	[5, 7]	4.70	11.40

1 derivatives (as e.g. generated by AD) against those computed by FD. In all runs this  
 2 test is successful as well. Thus the AD derivatives here prove to be a robust tool for  
 3 optimization. Moreover, they are efficient since they save about 20% cputime.

4 Using formulation (17) of the optimization problem the size of the problem becomes  
 5 much bigger since now  $\vec{m} \in \mathbb{R}^N$  is a variable, too. Therefore it is recommended to start  
 6 with a subset of the data points, and increase the number of data points successively.  
 7 We use every 100th point of the original data. Moreover, we exclude those where  
 8 the volume transport is close to zero ( $\vec{m}_j < 10^{-6}$ ). It turns out that also the part of  
 9 the upper branch of the hysteresis curve where the slope becomes too big (that is for  
 10  $F_1 > 0.125$ ) have to be excluded to obtain a good fit. We started using  $N = 63$  data  
 11 points. Later on we incorporate more and more data points, but this does not give any  
 12 improvement.

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

23 As can be seen from Fig. 5 the fits obtained during the optimization are very good  
 24 in the middle range where  $F_1 \in [-0.1, 0.06]$ . The left part of the hysteresis curve is not  
 25 fitted well, a fact that cannot be improved by adjusting the weights  $\omega_j$  and/or  $\delta_j$ . But  
 26 this area is not very interesting for further transient computations: It corresponds to a  
 27 climate state with strong fresh water export from the Atlantic which is not expected  
 28 to be reached in the wake of global warming. The present climate is fitted very well,  
 29 which is natural since the additional constraints enforce a good match here. The kink

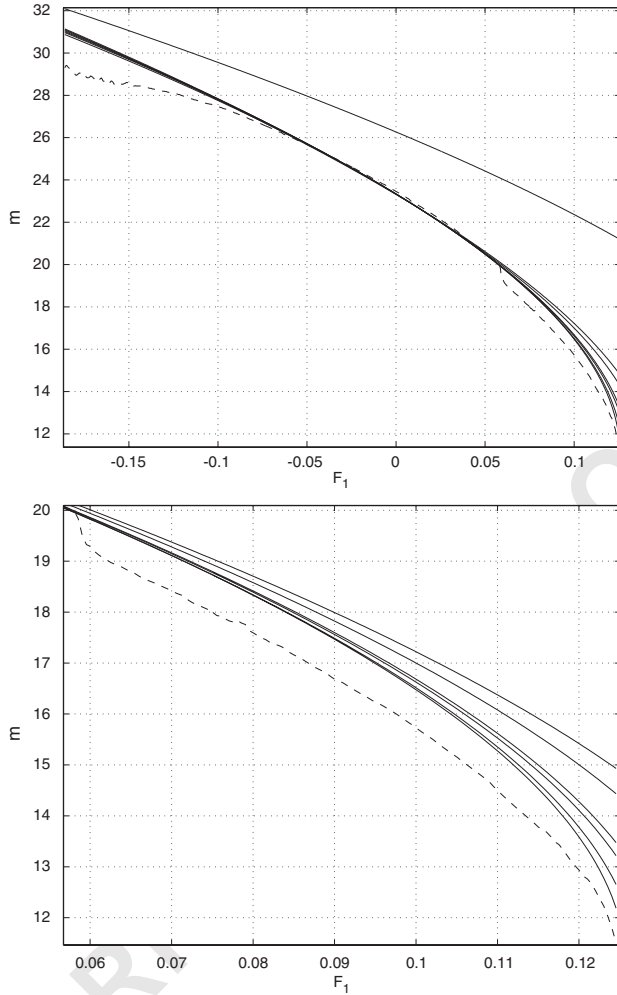


Fig. 5. Top: fitted curves of stationary volume transport  $m(F_1)$  for the box model. Dashed: data, solid: curves obtained with optimized parameters at the start and after 100, 200, 300, 400, 500, and 562 iterations (from top to bottom, compare Table 4), bottom: zoomed near the point of collapsing.

- 1 in the hysteresis curve at  $F_1 \approx 0.06$  cannot be fitted by the reduced-form model.  
 Moreover, there remains a discrepancy between the data and the fitted curve  
 3 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  
 5  $F_1 = 0.014 = F_e$  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  
 7 adjusting the weights  $\omega_j$  in this area do not lead to better results. Thus summarizing  
 the weights  $\omega_j, \delta_j$  in this examples were a useful method to test the limits of the model.

## 1 8. Summary

3 In this paper we perform a parameter optimization for a reduced-form model of  
4 the Atlantic thermohaline circulation. The output of the stationary model is fitted to  
5 data provided by quasi-stationary runs with CLIMBER-2, a climate model of intermedi-  
6 ate complexity. Additional constraints concerning the present climate state are imposed.  
7 We derive alternative equations characterizing the stationary state, show their equiva-  
8 lence, and discuss their differences with respect to numerical solution techniques. Here  
9 we use and compare available software in MATLAB. We discuss different formulations  
10 of the optimization problem and show how to solve it using an SQP method from  
11 MATLAB's Optimization Toolbox. For the computation of descent directions we use  
12 exact derivative information obtained by using a library for algorithmic or automatic  
13 differentiation (AD). The AD-generated derivatives show very good agreement with  
14 finite difference approximations. Moreover, they are successfully used in the optimiza-  
15 tion and lead to about 20% saving in cputime. The results of the optimization show  
16 a significant improvement in the fit of the reduced-form model compared to the initial  
17 parameter values, specifically in the most interesting area, namely the bifurcation point  
and the present-day state of the Atlantic thermohaline circulation.

## Acknowledgements

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.

## 19 References

- 21 [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, G. Bond, Evidence for general instability of past climate from a 250-kyr ice-core record, *Nature* 364 (1993) 218–220.
- 23 [2] A. Ganopolski, S. Rahmstorf, V. Petoukhov, M. Claussen, Simulation of modern and glacial climates with a coupled global model of intermediate complexity, *Nature* 391 (1998) 351–356.
- 25 [3] D. Herbison-Evans, Caterpillars and the inaccurate solution of cubic and quartic equations, *Australian Comput. Sci. Commun.* 5(1) (1983), see also [<http://www-staff.mcs.uts.edu.au/~don/pubs/solving.html>].
- 27 [4] INTEGRATION: Integrated assessment of changes in the thermohaline circulation [<http://www.pik-potsdam.de/~stefan/Projects/integration/index.html>].
- 29 [5] S. Manabe, R.J. Stouffer, Century-scale effects of increased atmospheric CO<sub>2</sub> on the ocean-atmosphere system, *Nature* 364 (1993) 215–218.
- 31 [6] W.D. Nordhaus, *Managing the Global Commons: The Economics of Climate Change*, MIT Press, Cambridge, MA, 1994.
- 33 [7] *Optimization Toolbox For Use with MATLAB User's Guide Version 2*, The Mathworks Inc., Natick, MA, USA, 2000.
- 35 [8] V. Petoukhov, A. Ganopolski, V. Brovkin, M. Claussen, A. Eliseev, C. Kubatzki, S. Rahmstorf, CLIMBER-2: a climate system model of intermediate complexity. Part I: model description and performance for present climate, *Climate Dyn.* 16 (2000) 1–17.
- 37

- 1 [9] S. Rahmstorf, Bifurcations of the Atlantic thermohaline circulation in response to changes in the  
hydrological cycle, *Nature* 378 (1995) 145–149.
- 3 [10] S. Rahmstorf, On the freshwater forcing and transport of the Atlantic thermohaline circulation, *Climate  
Dyn.* 12 (1996) 799–811.
- 5 [11] S. Rahmstorf, A. Ganopolski, Long-term global warming scenarios computed with an efficient coupled  
climate model, *Climatic Change* 43 (1999) 353–367.
- 7 [12] S.H. Schneider, S.L. Thompson, Simple climate model used in economic studies of global change, in:  
S.J. DeCanio, R.B. Howarth, A.H. Sanstad, S.H. Schneider, S.L. Thompson (Eds.), *New Directions in  
the Economics and Integrated assessment of Global Climate Change*, Pew Center on Global Climate  
9 Change, Arlington, VA, 2000, pp. 59–80.
- 11 [13] J.R. Scott, J. Marotzke, P.H. Stone, Interhemispheric thermohaline circulation in a coupled box model,  
*J. Phys. Oceanogr.* 29 (1999) 351–365.
- 13 [14] T.F. Stocker, A. Schmittner, Influence of CO<sub>2</sub> emission rates on the stability of the thermohaline  
circulation, *Nature* 388 (1997) 862–865.
- 15 [15] H. Stommel, Thermohaline convection with two stable regimes of flow, *Tellus* 13 (1961) 224–241.
- 17 [16] S. Titz, T. Kuhlbrodt, U. Feudel, S. Rahmstorf, On freshwater-dependent bifurcations in box models of  
the interhemispheric thermohaline circulation, *Tellus* 54 (1) (2002) 89–98.
- 19 [17] F.L. Toth, T. Bruckner, H.-M. Füssel, M. Leimbach, G. Petschel-Held, Integrated assessment of  
long-term climate policies: Part 1—Model presentation, *Climatic Change* (2003) forthcoming.
- 21 [18] A. Verma, ADMAT project homepage, Cornell University, Ithaca, NY, USA  
[<http://www.tc.cornell.edu/~averma/AD/ADMAT.html>].
- 23 [19] A. Verma, ADMAT: Automatic Differentiation in MATLAB using object oriented methods  
[<http://www.tc.cornell.edu/~averma/AD/admatoo.ps>].
- 25 [20] K. Zickfeld, S. Rahmstorf, T. Slawig, A reduced-form model for the response of the Atlantic  
thermohaline circulation to global warming, 2003, submitted for publication.