Technical University of Munich
Department of Mathematics

Impact of Parameter Uncertainties in Ice Sheet Models

Masters's Thesis by Ekin Su Köksal

Supervisor:	Prof. Dr. Christian Kühn
Advisors:	Prof. Dr. Christian Kühn and Dr. Kerstin Lux
Submission Date:	September 28th, 2022

I hereby declare that this Master's thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Dachau, September 28th, 2022

Zusammenfassung

Die vorliegende Masterarbeit untersucht Auswirkungen von Parameterunsicherheiten auf ein nichtlineares und ein-dimensionales Grönland Eisschild Modell. Dafür werden Sensitivitätsanalysen und probabilistische Analysen auf das Modell angewandt und zu jeder dieser Analysen wird ein numerisches Framework mittels `MATLAB` erstellt.

Zunächst werden unter anderem numerische Fortsetzungsmethoden auf das Grönland Eisschild Modell angewendet. Basierend auf den Entscheidungskriterien, wie Ergebnisqualität oder Performance, ist `MATCONT` dafür am geeignetsten.

Die Sensitivitätsanalyse umfasst die lokale Sensitivität, globale Sensitivität mit der Sobol Methode und die Sensitivität von Bifurkationskurven. Die Modellparameter besitzen eine Wahrscheinlichkeitsdichtefunktion, für das in dieser Arbeit betrachtete Modell wird eine uniforme Verteilung verwendet. Dabei liegt der Fokus auf der Sobol Methode, die während dieser Arbeit in `MATLAB` implementiert wird. Die Besonderheit des Codes ist, dass dieser allgemein gehalten ist und damit auch auf andere Klimamodelle mit beliebig vielen Parametern angewandt werden kann.

Um die Sensitivität von Bifurkationskurven zu untersuchen, wird die Wasserstein-Distanz von Bifurkationskurven berechnet und davon dann die Sensitivität mit der Sobol Methode bestimmt. Diese Vorgehensweise wird auf Beispielmodelle angewandt, um zunächst die Gültigkeit der Idee zu testen. Im Anschluss werden Verbesserungsvorschläge gegeben, um diesen Ansatz künftig weiterführen zu können.

Mit der probabilistischen Analyse wird ein analytischer und numerischer Weg zur Bestimmung der Wahrscheinlichkeitsdichtefunktion sowie der kumulierten Verteilungsfunktion der kritischen Erdoberflächentemperatur vorgestellt. Das Ergebnis ist dabei von den Wahrscheinlichkeitsdichtefunktionen der Modellparameter abhängig.

Die Arbeit umfasst nicht nur mathematische Herangehensweisen, sondern auch eine ausführliche Dokumentation des `MATLAB` Frameworks. Damit wird es Leser*innen ermöglicht, die Wahrscheinlichkeitsdichtefunktion der Parameter des Modells zu ändern oder sogar auf andere Klimamodelle anzuwenden.

Abstract

In this Master's thesis, the impacts of parameter uncertainties on a nonlinear and one-dimensional Greenland ice sheet model are investigated. To do this, sensitivity analyses and probabilistic analyses are applied to the model and a numerical framework for each of these analyses is constructed using `MATLAB`.

First, numerical continuation methods are applied to the Greenland ice sheet model. `MATCONT` is the most suitable for this purpose based on decision criteria, such as result quality or performance.

The sensitivity analysis includes local sensitivity, global sensitivity with the Sobol method and the sensitivity of branches of equilibria. The model parameters have a probability density function, and a uniform distribution is used for the model considered in this thesis. The focus is on the Sobol method, which is implemented in `MATLAB` in this thesis. The special feature of the code is that it is general and thus can be applied to other climate models with any number of parameters.

To examine the sensitivity of branches of equilibria, the Wasserstein distance of branches of equilibria is calculated and from this the sensitivity is determined using the Sobol method. This approach is applied to example models to first test the validity of the idea. Suggestions for improvement are then given to further develop this approach in the future.

Probabilistic analysis is presented as an analytical and numerical way to determine the probability density function as well as the cumulative distribution function of the critical earth surface temperature. The result depends on the probability density functions of the model parameters.

The thesis not only provides mathematical approaches, but also detailed documentation of the `MATLAB` framework. This allows the reader to modify the probability density function of the model parameters or even apply it to other climate models.

Contents

List of Figures

List of Tables

This page intentionally left blank

1. Introduction

Environmental and climatological changes are one of the great threats of our times. Knowing about the impacts of these changes is therefore of utmost importance. In the Intergovernmental Panel on Climate Change (IPCC) assessment reports, the "state of knowledge on climate change" is determined and topics that need more research in the future are identified. [25] Not only are climate assessment reports published, but also special reports such as "Global Warming of 1.5 °C" and "The Ocean and Cryosphere in a Changing Climate", to name a few. [25, 62]

Ice sheets and glaciers contribute to sea level rise as they lose mass due to enhanced warming of the climate. [62, A. 1.1, FAQ 1.1] Note that the Greenland ice sheet alone has lost $278 \pm 11\,\mathrm{Gt\,year^{-1}}$ of ice mass on average, which has caused the sea level to rise about $0.77 \pm 0.03\,\mathrm{mm\,year^{-1}}$. [62, A. 1.1]

The loss of ice mass of the Greenland ice sheet can be explained through a cycle of processes: High temperatures lead to the Greenland ice sheet losing more ice mass, which ensures that more sunlight can be absorbed and the average surface temperature of the Earth increases, causing more loss of ice mass in the Greenland ice sheet, and so on. [2, Fig. 16.8] [72] Eventually, a tipping point in the Earth system is passed such that an irreversible process is triggered and the Greenland ice sheet can no longer grow. [62, Sec. 4.2.3.5] [72]

Hence, research questions regarding the tipping point can be investigated with climate models. Often, in order to model processes in nature mathematically and physically, complexity is reduced and simpler assumptions are made. [2, Ch. 16, p.456] Ultimately, parameter uncertainties arise for these models, which are then often considered probabilistically. [69, Sec. 1.1] So, this brings us to the key goal of this thesis, which is to analyze the impacts of parameter uncertainties of a Greenland ice sheet model through sensitivity or probabilistic analyses. Each analysis is supported by a numerical framework implemented using `MATLAB`.

At this point, we would like to emphasize that this thesis combines mathematics and climate science. Yet, the impacts of parameter uncertainties are approached from a mathematical standpoint, leaving interpretations that fall within the realm of climate science to climate scientists.

The thesis is structured as follows: Chapter 2 begins with the basics of climatology. Important terms are introduced and put into context. Next, the mathematical basics, specifically the bifurcation theory is explained. Chapter 4 then introduces the

Greenland ice sheet model and analyzes it using numerical continuation. Thereby, different methods are compared. To examine how small variations in a model input affect the model output, Chapter 5 addresses local and global sensitivity analyses. Following, the bifurcation and sensitivity analysis is linked using Wasserstein-inspired as well as Wasserstein distances. Finally, in Chapter 7, the probability density function for the critical surface temperature is calculated analytically and numerically.

The two supervisors of this thesis are involved in the EU Horizon 2020 project Tipping Points in the Earth System (TiPES). [71] "The main object [of that project] is to better quantify the tipping elements that are present in the climate system and to ensure that climate projections also include these." [70] The subject of this thesis was established with the TiPES project in mind.

2. Climatological Fundamentals

This chapter collects basic topics necessary for understanding this work. It does not claim to be complete, but is only a supplement to explain the basics of climate to the mathematical reader.

The interested reader can gain more knowledge to each topic by reading the book "Meteorology today: an introduction to weather, climate, and the environment" by C. Donald Ahrens [2].

Sources are from C. Donald Ahrens [2] and the Intergovernmental Panel on Climate Change (IPCC) Special Report on the Ocean and Cryosphere in a Changing Climate [62]. Other sources are cited accordingly.

Ice Sheet, Glacier Ice sheets and glaciers cover 10 % of Earth's land area and hold over two thirds of Earth's freshwater. Their mass grows or shrinks depending on atmospheric changes. The two largest ice sheets are the Greenland and Antarctic ice sheets.

Dynamic Thinning The movement of ice sheets towards outlet glaciers or ice streams is called *ice sheet flow* which causes spreading and thinning of the ice sheet. [61] An increased flow rate is called *dynamic thinning*, i.e., "glacier flow acceleration". [62, p. 237]

Global Mean Surface Temperature The *global mean surface temperature (GMST)* is recorded since the year 1880. [4] Historic records show the increase of the average surface temperature began around 1900. In the year 2021, the average Earth's surface temperature was around 0.85°C. [4]

As we can see in Fig. 2.1, the global mean surface temperature (GMST) has increased by approximately 1°C since the recordings began. Although this may not seem "much" in general, it is still alarming since the Earth's surface temperature has increased by 0.5°C from 1880 to 2000 (120 years) and then again by approximately 0.5°C from 2000 to 2020 (20 years). This shows more and rapid increase of the GMST with progressing time. [4]

Please note, when speaking about the GMST, we refer to the temperature *above the pre-industrial level*. [34]

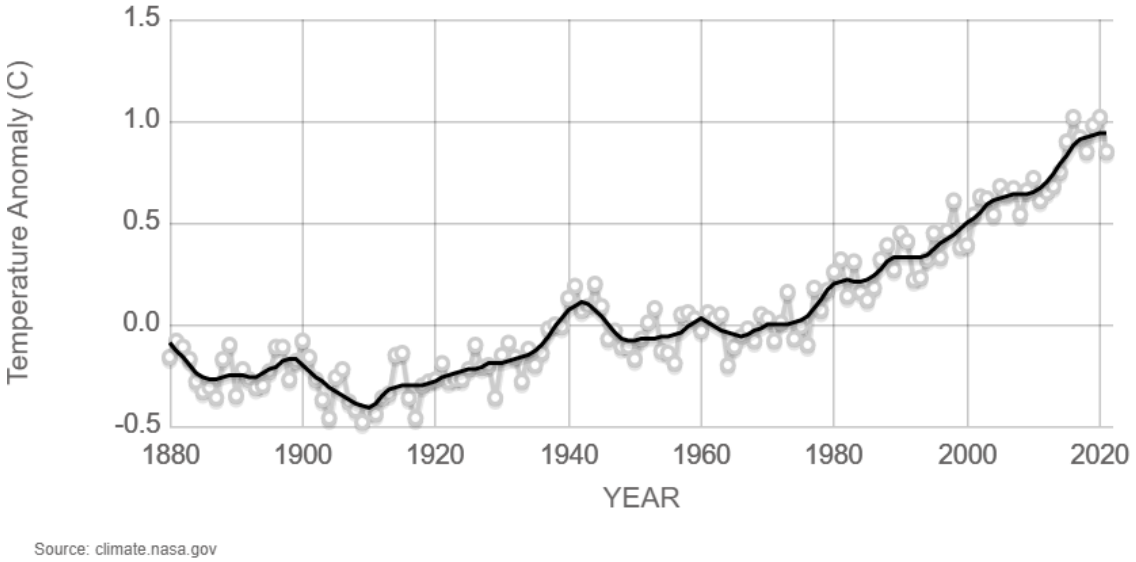


Figure 2.1: Global mean surface temperature above the pre-industrial level from 1880 to 2021. [4]

Greenland Ice Sheet The *Greenland ice sheet* gains and loses mass through snowfall and melting of ice shelves depending on processes in the atmosphere. Their difference is referred to as *surface mass balance (SMB)*. In the 1900s, mass gain and loss was in balance. Due to increasing GMST and increase in dynamic thinning, the SMB reduced and the Greenland ice sheet has lost more mass.

From 2006 to 2015, the global sea level has risen about $0.77 \pm 0.03\,\mathrm{mm\,year^{-1}}$, mostly as a result of surface melting of the Greenland ice sheet. If melted off completely, the sea level would rise by more than 7 m. [34]

The consequences of the complete melting of the Greenland ice sheet would be flooding, disruption of the overturning circulation of the ocean, and damages on coastal ecosystems. [34]

Tipping Point A *tipping point* can be defined as a "critical threshold at which a tiny perturbation can qualitatively alter the state or development of a system". [33, p. 1786]

The Working Groups I and II of the IPCC describe in "The Ocean and Cryosphere in a Changing Climate" a tipping point as "a critical threshold at which global or regional climate changes from one stable state to another stable state". [62, p. 699]

Albedo Albedo is defined as "the percent of radiation returning from a surface compared to that which strikes it". [2, p. G-1]

Feedback Mechanism A *feedback mechanism* is a cycle of implications that shows whether a process is strengthened (*positive feedback*) or weakened (*negative feedback*) by circumstances. In the climatological context, one can see if the surface temperature increases (or decreases) depending on processes affecting the Earth-atmosphere system. This means, that the surface temperature can be increased, if the flow of events cycles is being strengthened.

Some examples of feedback mechanisms are the water vapor-greenhouse feedback, ice-albedo feedback and cloud feedback. Thus, feedbacks can be used to simulate climate models more accurately. [24]

Ice-Albedo Feedback The increasing surface temperature ensures the melting off the snow and ice layer which decreases the albedo. Hence, more sunlight can be absorbed by the surface instead of being reflected back to space. This positive feedback mechanism is called *ice-albedo* or *snow-albedo feedback*. [24] We can observe the process of enhancing of the surface temperature in Fig. 2.2.

In [24], one can also notice that when the overlying atmosphere is taken into account, the ice-albedo effect becomes a negative feedback. However, this thesis will only consider the positive ice-albedo feedback.

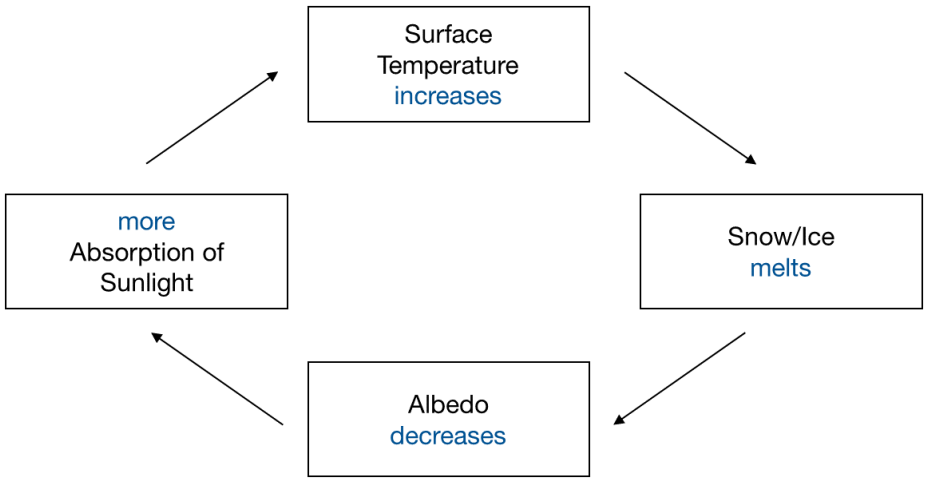


Figure 2.2: Positive Ice-Albedo Feedback, inspired by [2, Fig. 16.8] and [24, Fig. 7.6]. When following the arrows, one can see how the cycle of processes is being enhanced, thus the surface temperature getting warmer.

Melt-Elevation Feedback The reason the critical threshold of the Greenland ice sheet is triggered is the (positive) *melt-elevation feedback*. Due to increased GMST, the ice sheet melts and thus decreases in height. With the implications explained in the ice-albedo feedback, the surface temperature gets warmer and causes more melting of the ice sheet. [5]

Melting Sensitivity of Ice The *melting sensitivity of ice* γ is defined as "increase in surface melt rate per degree of warming". [34, p. 1800]

Atmospheric Lapse Rate The *atmospheric lapse rate* Γ denotes the rate at which the "temperature decreases linearly with the height of the ice surface". [34, p. 1800]

3. Mathematical Fundamentals

The complete melting of the Greenland ice sheet could lead to serious consequences, as we have discussed in Chapter 2. Although we can observe some fluctuations in the surface mass balance (SMB), such as the growth of the Jakobshavn glacier since 2016, these cannot stop the increasing melting of the Greenland ice sheet due to anthropogenic global warming in the long term. [5]

Ultimately, the melting of the Greenland ice sheet leads to surpassing a tipping point which triggers irreversible processes, like the Greenland ice sheet not growing again. [62] Further, exceeding the tipping point would cause the triggering of other tipping points, for example in the Amazon rainforest and the tropical monsoon systems and thus upsetting the balance of the Earth. [2, 5]

That is why it is important to analyze how certain preconditions affect the tipping point which corresponds to the surface temperature of the Earth.

As described in Chapter 2, a tipping point can change the quality of a system and cause it to lose its stability. From a mathematical point of view, tipping points of a system can be described by bifurcations. [35] This is because tipping points "are often caused by a change in the topological appearance of the system's phase portrait upon parameter variation". [35, p. 2]

Consequently, we will cover fundamentals of ordinary differential equations (ODEs), bifurcations and numerical continuation.

3.1 Introduction to Ordinary Differential Equations

We introduce the theory of parameter dependent ordinary differential equations (ODEs) as it forms the basis for the bifurcation theory in Section 3.2 and hence for the analysis of the tipping points.

Definition 3.1 (Ordinary Differential Equation [8, Sec. 1.1])**.** *Let $\mathcal{T} \subseteq \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^n$ and $\Lambda \subseteq \mathbb{R}^k$ be open subsets. An* ordinary differential equation (ODE) *that models how a physical process evolves over time is defined as*

$$\frac{dx(t)}{dt} \equiv x' = f(t, x, \lambda), \tag{3.1}$$

where $f : \mathcal{T} \times \mathcal{X} \times \Lambda \to \mathbb{R}^n$ *is a (nonlinear) continuously differentiable function,* $x \in \mathcal{X}$ *is the vector of state variables,* $t \in \mathcal{T}$ *is the independent variable time, and* $\lambda \in \Lambda$ *is the vector of parameters.*

Remark 3.1 (Smooth Function [8, Sec. 1.1])**.** A function that is continuously differentiable is also called a *smooth* function.

To address questions of existence and uniqueness of solutions, we define an initial value problem (IVP).

Definition 3.2 (Initial Value Problem [8, Sec. 1.1])**.** *Let* $(t_0, x_0) \in \mathcal{T} \times \mathcal{X}$*. Consider an ODE as defined in Definition 3.1 Then, the pair*

$$\frac{dx(t)}{dt} \equiv x' = f(t, x, \lambda), \quad x(t_0) = x_0, \tag{3.2}$$

is called an initial value problem (IVP) *and the equation* $x(t_0) = x_0$ *is called an* initial value.

Consider an IVP as stated in Definition 3.2. Let be $\lambda_0 \in \Lambda_0$. The *Existence and Uniqueness Theorem* as described in Theorem 1.2 of C. Chicone can be applied to the IVP. This theorem is one of the fundamental issues of the theory of ODEs. For detailed information, refer to Section 1.1 of C. Chicone. [8]

We introduce a classification of ODEs as they are relevant for later results. Thus, from now on, we consider all ODEs to be autonomous ODEs as specified in Definition 3.3.

Definition 3.3 (Autonomous ODE [8, Sec. 1.2])**.** *An* autonomous ODE *is a type of ODE that is defined as*

$$\frac{dx}{dt} \equiv x' = f(x, \lambda), \tag{3.3}$$

where the function f *is* not explicitly *dependent on the independent variable.*

The solution curve of an autonomous ODE is called a trajectory. [75, Sec. 3.V]

Definition 3.4 (Phase Portrait [75, Sec. 3.V], [8, Sec. 1.3])**.** *The geometric representation of all trajectories of an autonomous differential equation in a phase plane is called a* phase portrait*.*

We would like to now discuss the stability of their equilibria x_*.

Definition 3.5 (Equilibrium, Steady State, Critical Point [8, Sec. 1.3])**.** *If* $f(x_*, \lambda) = 0$ *holds, then* x_* *is called an* equilibrium, steady state *or* critical point*.*

Stability theory analyzes the behavior of the solution or steady state of an autonomous ODE in the long-term despite small changes. Roughly speaking, if the solution or steady state remains in the neighborhood of the trajectory or equilibrium, then it is *stable*, otherwise *unstable*. [8, Fig. 1.8, Def. 1.38, Def. 1.39]

Definition 3.6 (Notions of Stability [20, Sec. 1.5, Definition 1.5.1, p. 24])**.** *Consider a nonlinear ODE $x' = f(x)$, where $x \in \mathbb{R}^n$ and $f(x) \in \mathbb{R}^n$.*

1. *"An equilibrium solution x_0 of [the ODE] is* stable *if for each $\epsilon > 0$ there exists a $\delta > 0$ such that if $x(t)$ $(t \geq 0)$ is a solution to [the ODE] with $\|x(0) - x_0\| \leq \delta$, then $\|x(t) - x_0\| \leq \epsilon$ for all $t \geq 0$."*

2. *"An equilibrium solution x_0 to [the ODE] is* asymptotically stable *if it is stable and furthermore there exists a $\delta \geq 0$ such that if $x(t)$ is a solution to [the ODE] for $t \geq$ ß and $\|x(0) - x_0\| \leq \delta$, then $x(t) \to x_0$ for $t \to \infty$."*

3. *"A solution to [the ODE] is* unstable *if it is not stable."*

3.2 Bifurcation Theory

We proceed with the bifurcation theory of one-dimensional ODEs to analyze their behavior qualitatively. [32, Sec. 2.1] Given an ODE as in Definition 3.3, consider the problem

$$f(x, \lambda) = 0, \tag{3.4}$$

where $f : \mathbb{R}^{k+1} \to \mathbb{R}$ is smooth. The key point of bifurcation theory is analyzing the stability of the equilibria under variation of the bifurcation parameter λ. [32, Sec. 2.1] [73]

Definition 3.7 (Topological Equivalence of Systems [32, Def. 2.1])**.** *Two systems are* topologically equivalent *to each other, if there exists a homeomorphism $h : \mathbb{R}^n \to \mathbb{R}^n$ that maps trajectories from one system to another, preserving the direction of time.*

This leads to the definition of a bifurcation.

Definition 3.8 (Bifurcation [32, Def. 2.11, p. 57])**.** *"The appearance of a topologically nonequivalent phase portrait under variation of parameters is called a* bifurcation*."*

"Thus, a bifurcation is a change of the topological type of the system as its parameters pass through a bifurcation (critical) value." [32, p. 57] We will use bifurcation diagrams to analyze the qualitative behavior of phase portraits. [8, 20]

Proposition 3.1 (Bifurcation Diagram [8, Sec. 1.3])**.** *A diagram that collects all phase portraits for different values of the parameter λ of the differential equation as defined in Definition 3.3 is called* bifurcation diagram*.*

We now consider *saddle-node bifurcations*, also called *fold bifurcations* as they are an important part of this thesis. [66] The general theorem of the saddle-node bifurcation will be given after first examining a simple exemplary ODE and its characteristics.

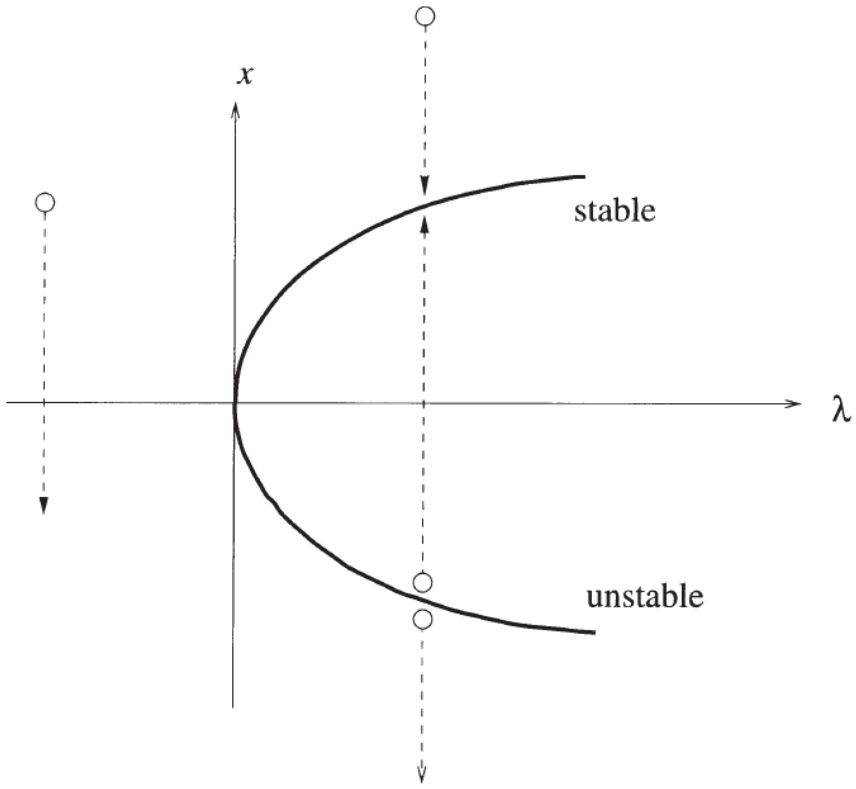


Figure 3.1: Example of a saddle-node bifurcation. Visualization of the branches of equilibria and their stability for $x' = \lambda - x^2$. [66, Fig. 2.1]

Example 3.1. [32, Sec. 3.2] [66] Consider the one-dimensional system

$$x' = \lambda - x^2. \tag{3.5}$$

Let us compute and analyze the equilibria of (3.5) for different values of λ. This shows that for $\lambda = 0$ there is an equilibrium at $x_* = 0$. For $\lambda < 0$, there exist no real equilibria. However, for $\lambda > 0$, the system has two fixed points $x_\pm = \pm\sqrt{\lambda}$. Based on A. Spencer and I. G. Graham [66] and Definition 3.6, the steady state $x_- = -\sqrt{\lambda}$ is unstable and $x_+ = \sqrt{\lambda}$ is stable.

This brings us to the characteristic of saddle-node bifurcations:

Proposition 3.2. *[32, Sec. 3.1, Sec. 3.2] In a saddle-node bifurcation, parameter variation leads either to a stable equilibrium and an unstable equilibrium, or to their collision to an equilibrium, or to their disappearance.*

A visualization of Example 3.1 is shown in Fig. 3.1. This type of figure is a *bifurcation diagram*. Thereby, the two solid curves for positive and negative values of x are called *branches of equilibria*. [21, Sec. 3.1] The point $(x_*, \lambda_*) = (0, 0)$ at which the branches of equilibria meet is called *bifurcation point*, *fold point*, *turning point*, or *saddle-node*. [21, Sec. 3.1] [66]

Let us give a general theorem to saddle-node bifurcations.

Theorem 3.1 (Topological Normal Form of the Saddle-Node Bifurcation [32, Sec. 3.3])**.** *Consider the one-dimensional ODE as given in Definition 3.3. If the conditions*

1. $f(0,0) = 0,\ f_x(0,0) = 0,$
2. $f_{xx}(0,0) \neq 0,\ f_\lambda(0,0) \neq 0,$

are satisfied, then the ODE is locally topologically equivalent near the origin $(x, \lambda) = (0,0)$ *to the normal form*

$$x' = \lambda \pm x^2, \tag{3.6}$$

which is also called topological normal form of the saddle-node bifurcation*.*

The proof to this theorem can be found in Y. A. Kuznetsov. [32, Sec. 3.3]

3.3 Introduction to Numerical Continuation

After having introduced the theory of bifurcation analysis, we now discuss briefly how to numerically compute and numerically visualize the branches of equilibria of one-dimensional ODEs (stated in Definition 3.3).

Numerical continuation is a technique that uses a suitable algorithm to find equilibria of the problem (3.4) introduced in the beginning of Section 3.2. [20, Sec. 2.3] Thereby, discrete and consecutive points are computed, which approximate the branches of equilibria. [19] Interpolating and visualizing these with a graphics package then gives the bifurcation diagram. [13, 66]

There exist toolboxes and methods to perform numerical continuation. Advantages of using numerical continuation are practicability, efficiency and the applicability on complex models. [13, 18]

Consider now Example 3.1. We want to apply numerical continuation to this problem. To do this, we use the `MATLAB` (`Version R2022a`) function `vpasolve` which "solve[s] symbolic equations numerically". [59] It is available in the Symbolic Math Toolbox. [59]

Hence, to find a consecutive sequence of points that approximates the branches of equilibria, we implement a `for`-loop that iterates over the bifurcation parameter λ. In each iteration, `vpasolve` solves Example 3.1 for a $\lambda \in [0,1]$ and stores the found equilibria in a solution vector.

A relevant part of the implementation is as follows

```
syms x;
S = vpasolve(-x.^2 + L(i) == 0, x);
```

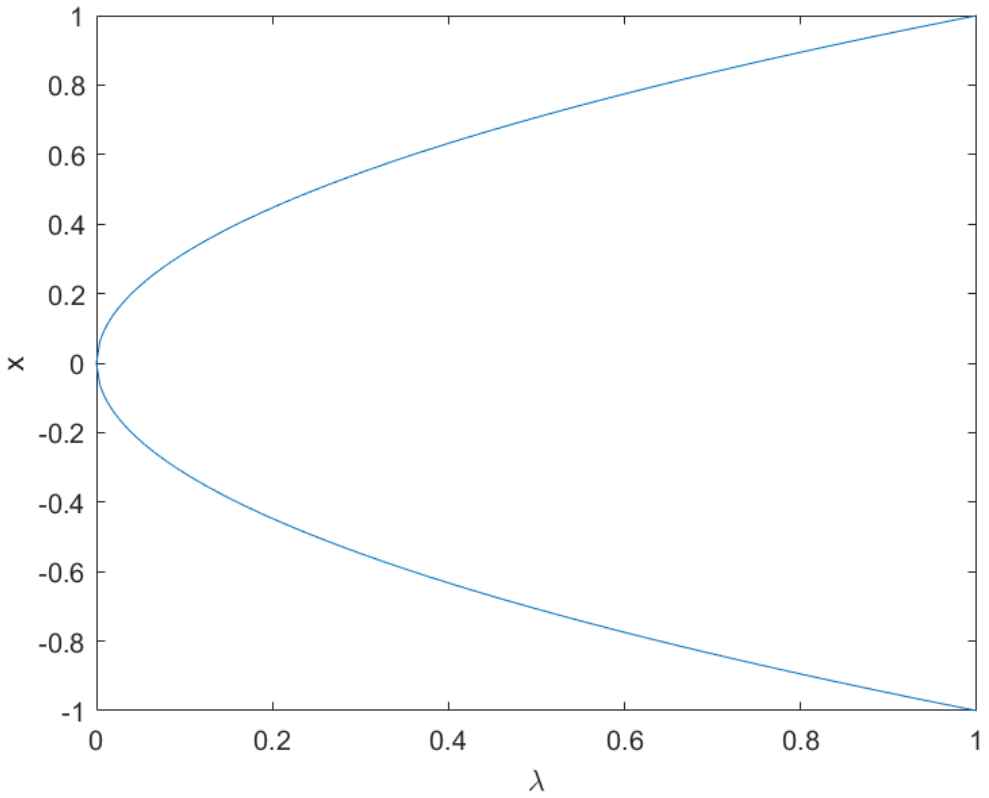


Figure 3.2: Bifurcation diagram of $x' = \lambda - x^2$ using `vpasolve`.

Hereby, this variable is defined as a *symbolic variable*. [59] The complete code can be found in the Appendix A.1. The resulting bifurcation diagram of Example 3.1 is achieved by interpolating the points from the solution vector (see Fig. 3.2). The blue solid lines are the branches of equilibria. Given the similarity of this diagram to Fig. 3.1, the approach of using `vpasolve` to numerically determining the bifurcation diagram of ODEs is confirmed.

Note, this section is only an outlook. In Section 4.2, we will see that `vpasolve` is not the only `MATLAB` function or package that can be used for numerical continuation.

4. Greenland Ice Sheet Model

Passing a critical tipping point would trigger an irreversible process of complete melting of the Greenland ice sheet. Such a tipping point can exist due to positive melt-elevation feedback. With increasing surface temperature, the thickness of the ice decreases, leading to potential vanishing of the Greenland ice sheet. [5]

We introduce a climate model that incorporates these features. To analyze its tipping points and start building the framework, different numerical continuation methods are applied to this model.

4.1 Model Introduction

In the publication "A simple equation for the melt elevation feedback of ice sheets" by A. Levermann and R. Winkelmann, an one-dimensional nonlinear climate model is introduced. This model is a time-evolution of the ice thickness of the Greenland ice sheet depending on the increasing surface temperature. [34]

Hereby, the connection between the ice thickness and surface temperature represents a self-enforcing feedback. The underlying assumptions are as follows: the "surface melt rate depends linearly on the surface temperature and [...] the temperature decreases linearly with the height of ice surface following a constant atmospheric lapse rate". [34, p. 1800]

The model is then derived from the Vialov profile which is "the mass conservation equation in the shallow ice approximation for flat glacier bed" [15, p. 887] and is given as [34]

$$\tilde{h}(x) = h_m\Big(1 - \big(\frac{x}{L}\big)^{\frac{n+1}{n}}\Big)^{(n/(2n+1))} \tag{4.1}$$

Hereby, h_m corresponds to maximum surface elevation, x is the horizontal position and L represents the horizontal limit of the ice sheet. Further, n is Glen's flow law exponent. [34] In our situation, the exponent is $\tilde{n} = 3$ since we consider ice flow in glaciers. [15, 34]

Notice, comparing the exponents in the Vialov profile (4.1) and Equation (4) in A. Levermann and R. Winkelmann yields $m = 2(\tilde{n} + 1) = 8$. [34]

Starting from the Vialov profile and using further derivations explained in Section 2.1 by A. Levermann and R. Winkelmann, we get the Greenland ice sheet model. [34]

Model 4.1 (Greenland Ice Sheet Model [34])**.** *The time-evolution of the ice thickness of the Greenland ice sheet depending on the temperature is given by the climate model*

$$\frac{dh}{dt} = -h^8 + \gamma\Gamma h - T. \tag{4.2}$$

The parameters are the atmospheric lapse rate $\Gamma = 5 \pm 2\,°C\,\mathrm{km}^{-1}$ *and the melting sensitivity of ice* $\gamma = 4.4 \pm 2\mathrm{cm}\,year^{-1}\,°\mathrm{C}^{-1}$*. The ice thickness is given by* h *and* T *is the surface temperature.*

For simplicity, the units of the parameters are omitted throughout the thesis.

4.2 Methods for Numerical Continuation

Numerical continuation is applied to the Greenland ice sheet model (Model 4.1). Thereby, we compare the `MATLAB` functions `vpasolve` and `fsolve`, and also the `MATLAB` software `MATCONT` based on algorithms, implementation, benchmarking results and limitations.

Preliminaries For better comparison of the numerical continuation methods, the parameters of Model 4.1 are set to the mid-point of the parameter ranges $\gamma = 440$ and $\Gamma = 0.005$. Recall, that the bifurcation parameter is T.

Information about Benchmarking The benchmarks are performed on a `Windows 11 21H2` machine with an 11th generation Intel i7 processor with 2.80 GHz and 16 GB of RAM. In terms of software configuration, the `MATLAB` function `timeit` is used for benchmarking. "In order to perform a robust measurement, [it] calls the specified function multiple times and returns the median of the measurements." [57]

4.2.1 `vpasolve`

In Section 3.3, numerical continuation was introduced using the `MATLAB` function `vpasolve`. In the following section, we solve the Greenland ice sheet model (Model 4.1) numerically with `vpasolve` and visualize the corresponding bifurcation diagram. For this purpose, we first start with the implementation. After that, we look into benchmarking and possible obstacles encountered. For the complete `MATLAB` code, see Appendix A.1.

Implementation The basic idea behind the implementation is similar as presented in Section 3.3. Solving the Greenland ice sheet model (Model 4.1) for the fixed parameter values $\gamma = 440$ and $\Gamma = 0.005$ while varying T, where $T \in [0, 5]$, results in the equilibria h_1, $h_2, \ldots, h_8$, where $h_i \in \mathbb{R}$ for $i = 1, 2, \ldots 8$. Intuitively, the ice thickness and surface temperature of the Earth are in $\mathbb{R}_+$. Hence, the search

interval is restricted to $[0, \infty]$. The MATLAB command for that part is implemented as follows:

```
syms h;
S = vpasolve(-h.^8+gamma*Gamma*h-T(i)==0, h, [0 Inf]);
```

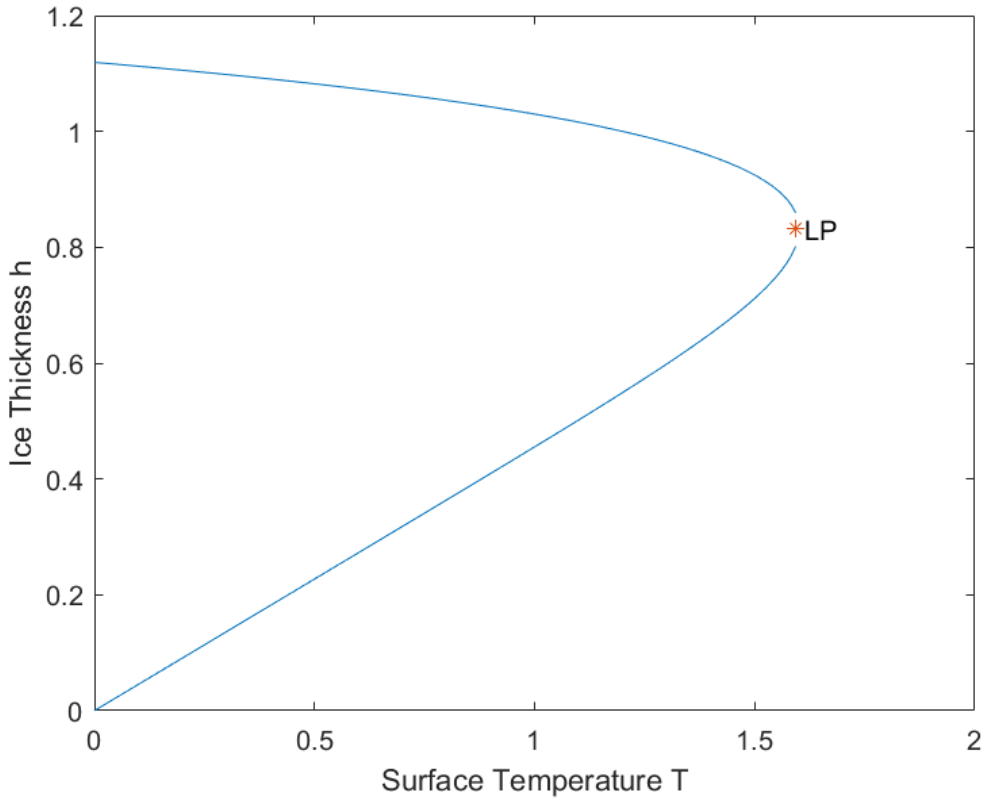
Simultaneously in each iteration step, the current solution T is noted separately. The reason for this implementation step is to find and mark the critical surface temperature T_c. The corresponding critical ice thickness h_c is calculated using Equation (9) in [34]. These points (T_c, h_c) are the tipping point, or limit point (LP), which is depicted as the orange star in Figures 4.1a and 4.1b.

Model 4.1 is then evaluated at 500 points for the parameter values $\gamma = 440$, $\Gamma = 0.005$, and $T \in [0, 5]$. The result can be seen in Fig. 4.1a).

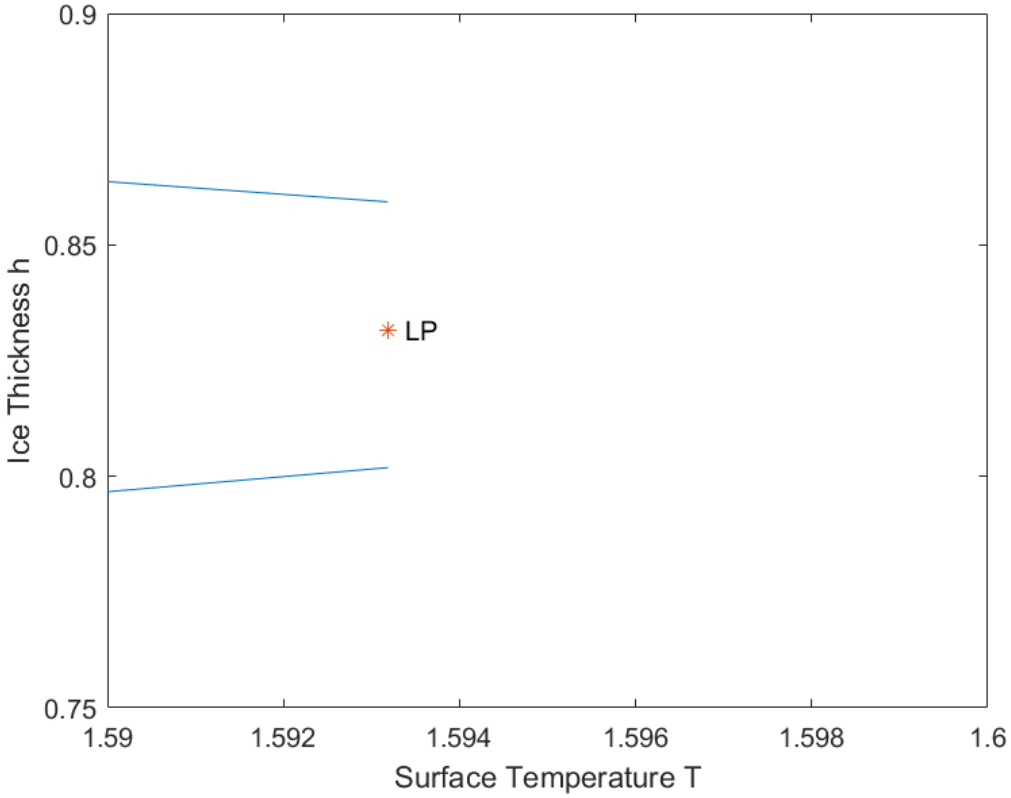
Benchmark Analysis A benchmark analysis is done to analyze the performance of the code. Therefore, we run the function `vpasolve_bifurcation` to numerically compute the bifurcation points of certain parameters γ and Γ (without the plotting part) using the MATLAB function `timeit`. It shows the median elapsed time for the code execution. [45, 57] The result is 12.6768 seconds.

Limitations The bifurcation diagram Fig. 4.1a shows a gap in the neighborhood around the critical point LP. If zoomed in (see Fig. 4.1b), we can observe that the critical point LP was placed in the center of the gap. But this is not necessarily the exact bifurcation point.

Even though the MATLAB function `vpasolve` has no problems with solving and visualizing Example 3.1, it has struggles with solving the Greenland ice sheet model (Model 4.1). Consequently, instead of this method, other algorithms should be used to achieve more accurate results. That is why in the subsequent section we will apply another MATLAB function that might overcome this problem.



(a) Bifurcation diagram using `vpasolve`.



(b) Zoom in on the bifurcation point.

Figure 4.1: Bifurcation diagram of the Greenland ice sheet model (Model 4.1) for the parameter values $\gamma = 440$ and $\Gamma = 0.005$ by applying the `MATLAB` function `vpasolve`.

4.2.2 `fsolve`

Next, we take a look at the `MATLAB` function `fsolve`. It is a root-finding algorithm that uses the Newton-based method *Trust-Region-Dogleg algorithm* [38, 40]. Such an algorithm is an iterative method which finds the minimum x_* of an objective function by approximating a model in a certain neighbourhood. [9, Sec. 1.1]

Same as the function `vpasolve`, this method is not specifically intended for solving bifurcation problems numerically, but can still be used as such. Further, the implementation principle is the same as `vpasolve`. The complete implementation can be found in the Appendix A.1.

Trust-Region-Dogleg Algorithm Without going into much detail, let us explain the basic working principle of such trust-region algorithms. At each step of the iteration, a *trust-region* is constructed around the iterate x_k by approximating f in the trust-region of x_k. A *trust-region* is a region "where we trust the model to be a faithful representation of the objective function". [9, p. 2] This trust-region contains a set of points that are in *trust-region radius* to x_k. If the difference between the approximation in step $k-1$ and k is sufficiently positive, then in the next iteration step $k+1$, the trust-region radius and hence the trust-region is made smaller. [9, Sec. 1.1]

The *trust-region-dogleg algorithm* is one of the most simple and widely used method which joins the iterates x_k with linear segments. [9, Sec. 1.1] This algorithm solves the *trust-region subproblem* [38, 78]

$$\min_d \big(\frac{1}{2} f(x_k)^\top f(x_k) + d^\top J(x_k)^\top f(x_k) + \frac{1}{2} d^\top f(x_k)^\top J(x_k) d\big), \text{ s.t. } \|Dd\| \leq \Delta. \quad (4.3)$$

Thereby, d is the search direction, D the diagonal scaling matrix, and Δ a positive scalar. [38]

Implementation The `MATLAB` function `fsolve` solves a nonlinear system denoted by $f(x) = 0$ for x. [40] Since Model 4.1 is also nonlinear, this `MATLAB` function can be applied to this model. The arguments of `fsolve` are the function $f(x)$ that is to be solved, an initial point x_0 and options, with the last one being optional. [40]

The first argument is implemented in `MATLAB` through an anonymous function set in $\mathbb{R}$:

```
gis_model = @(h)[real(-h.^8+Gamma*gamma*h-T)];
```

The initial points were selected as $x_0 = (0.1, 1)^\top$. Furthermore, the options were set to

```
options = optimoptions(@fsolve,'StepTolerance',1e-20,
  'FunctionTolerance',1e-20,'MaxFunctionEvaluations',
  1e5, 'MaxIterations',1e5,'Display','iter');
```

whereby the step tolerance was set from the default value 10^{-6} to 10^{-20}, the termination tolerance was set from 10^{-6} to 10^{-20}, the maximum number of function evaluations was set from 100 to 10^5, and the maximum number of iterations was set from the default value 400 to 10^5. [40]

Again, the `MATLAB` code iterates over $T \in [0, 5]$ and is evaluated at 500 points. In each iteration step, the equilibria h for the corresponding T are numerically computed using `fsolve` with the above arguments.

The result for the parameter values $\gamma = 440$ and $\Gamma = 0.005$ with the initial values $(0.1, 1)^\top$ can be seen in Fig. 4.2a. We observe that in comparison to the computation using `vpasolve`, all bifurcation points could be found as there are no gaps.

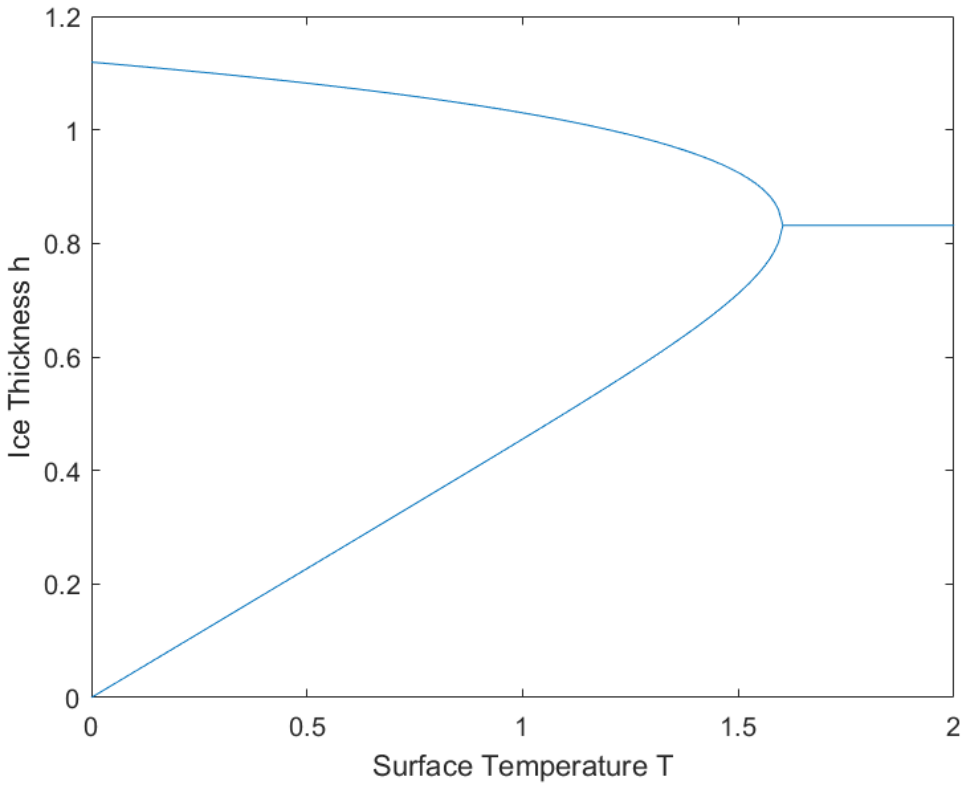
The plotting settings, number of iteration points and also the way of finding the bifurcation point LP are all carried out in the same way as with `vpasolve` (see Section 4.2.1).

Benchmark Analysis Benchmarking analysis shows that the elapsed time for the code execution with the intial points being $(0.1, 1)^\top$ is 6.8019 seconds. This solution is quite an improvement in comparison to `vpasolve`.

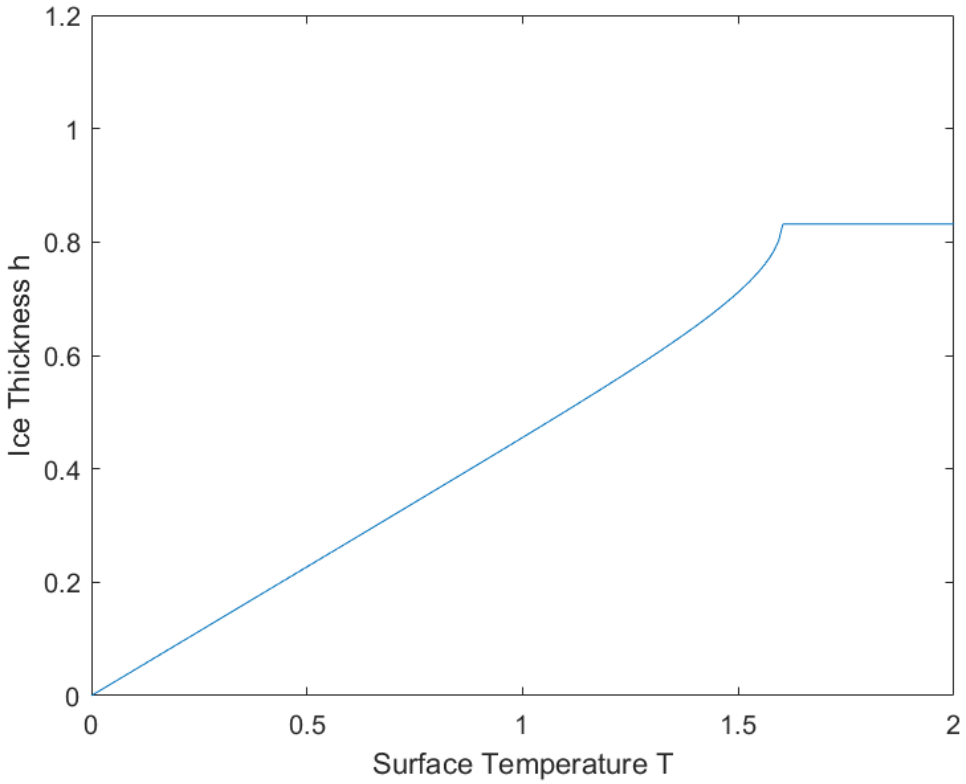
Limitations What stands out visually is the horizontal line at in Fig. 4.2a, starting at $T \approx 1.6$. These are the points at which the algorithm `fsolve` could not find a solution. In other words, the algorithm does not terminate after having computed the bifurcation points and arriving at the critical temperature T_c. It still continues until the limit for T is reached, which was set to 5 in our case. That is also why both of the plots (Fig. 4.2a and Fig. 4.2b) have no marked LP points.

This part may be avoided by integrating a code-snippet such that the code terminates if the solution for a specific T cannot be computed. Furthermore, choosing the initial points $(0, 0)^\top$ leads to only one branch of equilibria, as we can see in Fig. 4.2b. Thus, choosing the right initial point is important for this algorithm.

All in all, even if some obstacles of numerical continuation using `vpasolve` were avoided with `fsolve`, there are still small hiccups.



(a) Bifurcation diagram with the initial values $(0.1, 1)^\top$.



(b) Bifurcation diagram with the initial values $(0, 0)^T$.

Figure 4.2: Bifurcation diagrams of the Greenland ice sheet model (Model 4.1) for the parameter values $\gamma = 440$ and $\Gamma = 0.005$ and different starting points obtained with the `MATLAB` function `fsolve`.

4.2.3 MATCONT

In this section, we consider a numerical continuation and bifurcation toolbox for MATLAB, which is MATCONT. It is a GUI-MATLAB package, but the functions defined in this package can also be used directly. [19] MATCONT can be downloaded from [60] and as of this writing the last version is from July 2021 (Version 7p3).

In the following, we present the algorithm that MATCONT uses for numerical continuation and apply it to the Greenland ice sheet model (Model 4.1).

Pseudo-Arclength Continuation A. Spence and I. G. Graham point out in [66] that Newton's method has difficulties in producing results in the neighbourhood of the bifurcation point of an ODE system because the Jacobian matrix becomes singular.

The numerical continuation method *pseudo-arclength continuation* can, however, overcome this problem. It is a *predictor-corrector* algorithm and consists of the steps *prediction* and *correction*. [66] Fig. 4.3 illustrates the steps of the pseudo-arclength continuation method.

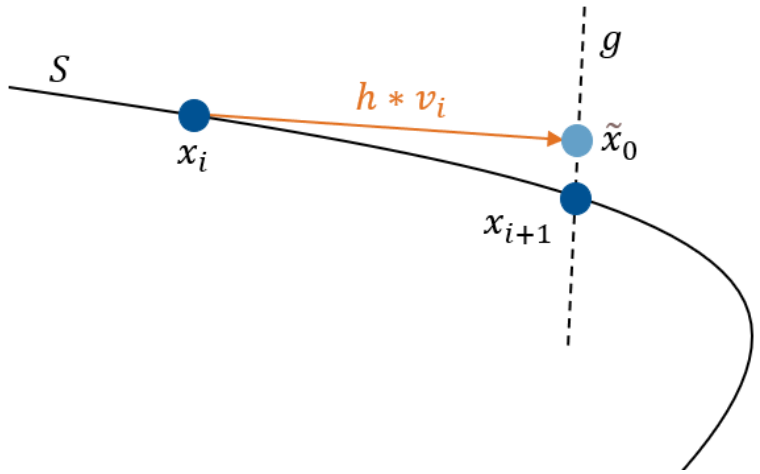


Figure 4.3: Illustration of the pseudo-arclength continuation method based on [66] and [19].

Consider that the points $(x, \lambda) \in \mathbb{R}^{k+1}$ interpolating the bifurcation diagram are points on the bifurcation curve, and thus are points of an arc S. [66] Let us assume that there is a point x_i on the curve with the tangent vector v_i fulfilling $f_x(x_i)v_i = 0$, where $\langle v_i, v_i \rangle = 1$. Here, $i = 1, 2, \ldots$ is the number of points on S. [19]

The first step is the *tangent prediction*: From x_i, a new point

$$\tilde{x}_0 = x_i + hv_i, \tag{4.4}$$

with $h > 0$ being the stepsize is predicted. This point $\tilde{x}_0$ is on a hyperplane $g(x) = \langle x - \tilde{x}_0, v_i \rangle$ orthogonal to the vector v_i . [19, 66]

The second step is *correction*: The predicted point $\tilde{x}_0$ is corrected and thus projected onto the curve S. To do so, apply Newton's method to $\tilde{x}_0$ [19, 66]

$$\tilde{x}_{k+1} = \tilde{x}_j - H_x^{-1}(\tilde{x}_j)H(\tilde{x}_j), \quad j = 1, 2, \ldots, \tag{4.5}$$

which converges to the new point x_{i+1}. Hereby, $H(\tilde{x}) = (f(\tilde{x}), 0)^\top$ and $H_x(\tilde{x}) = (f_x(\tilde{x}), v_i^\top)^\top$. [19]

Lastly, the tangent vector of the newly found point x_{i+1} should fulfil the following conditions: $f_x(_{i+1})v_{i+1} = 0$, and also [19]

$$\begin{pmatrix} f_x(x_{i+1}) \\ v_i^\top \end{pmatrix} v_{i+1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{4.6}$$

After having found the next approximated point x_{i+1} with the corresponding tangent vector v_{i+1}, one has to normalize this vector and apply the predictor-corrector steps to x_{i+1}. Meanwhile, there are several aspects to consider: [19]

- tolerance criterion: $\|f(x_i)\| \leq \epsilon$ for some $\epsilon > 0$
- accuracy condition: $\|\delta x_i\| \leq \tilde{\epsilon}$ for some $\tilde{\epsilon}$ and δx_i being the last point of the correction step
- stepsize h: right choice of h to not miss details of the bifurcation diagram

The algorithm is called *pseudo-arclength* because the points of the bifurcation are approximated, hence *pseudo*, on an arc with an arc length as the distance to the next point. [12, 66]

Implementation The idea is to implement a `MATLAB` function that numerically computes the bifurcation points as well as the corresponding tangent vectors and indicates the location and type of the bifurcation point for our Model 4.1. This function is `bifurcation_points` and has two inputs: γ and Γ, which are scalars. The code can be found in Lst. 4.1.

The Model 4.1 has the three parameters γ (`gamma`), Γ (`Gamma`), and T (`T`), while the last one is the bifurcation parameter. Since we know that at $(T_*, h_*) = (0, 0)$ is an equilibrium, we set the initial point for the temperature to be 0 (line 2). Furthermore, the bifurcation parameter is the active parameter, which is why we set `ap = 3` (line 3).

The initial tangent vector v_0 of x_0 is optional. If it is not specified, then this will be computed by `MATCONT`. [19] Then, the initial points are computed via `init_EP_EP` (`MATCONT` function) in line 4. Its first input parameter is our model. This is created using the `GUI`. [19] Fig. 4.4 shows how the ODE is introduced.

The maximum number of points on the curve were set to 100 and by marking the singularities as 1, it was flagged that our problem has a singular matrix. [19]
To perform a numerical continuation, the function `cont` from `MATCONT` is executed in line 9 of Lst. 4.1. It has several input arguments: `@equilibrium`, initial points `x0` and `v0`, and settings. [19]

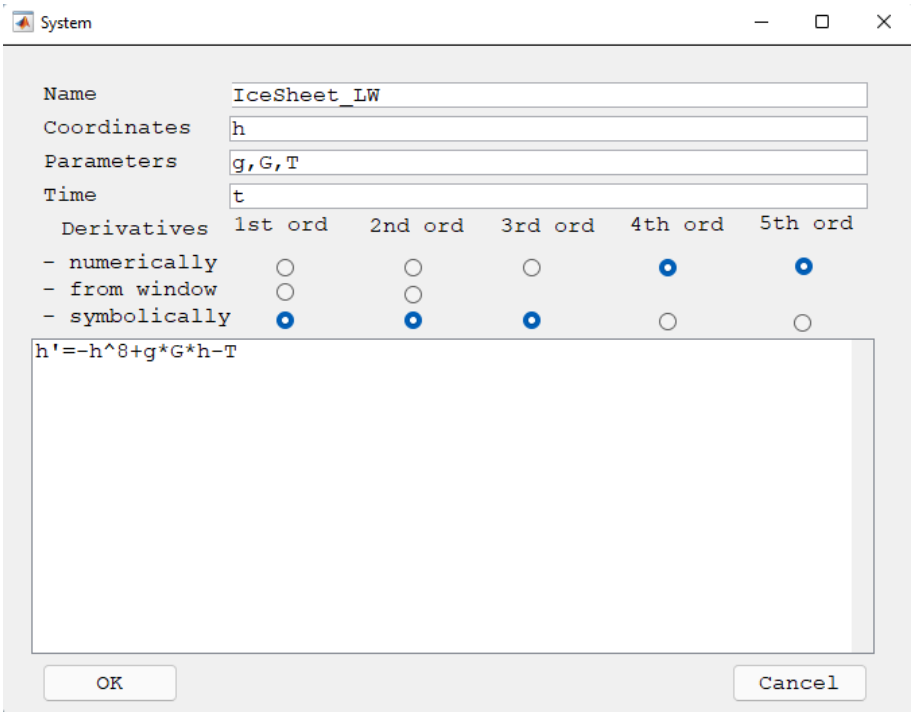


Figure 4.4: System definition, similar to [19, Fig. 5].

.

By calling the function `@equilibrium`, we specify that we want to compute the equilibria of our problem. [19]

In order to be able to continue computing with the results in later analyses, we store them in a structured array (`MATLAB` function `struct`). [56] It contains the parameter values (`gamma` and `Gamma`), points of the curve (`x` and `v`), and also information about found singularities (`s`).

```
1  function bfpts = bifurcation_points(gamma, Gamma)
2      p = [gamma; Gamma; 0];
3      ap = 3;
4      [x0,v0] = init_EP_EP(@IceSheet_LW,0,p,ap);
5      opt = contset;
6      opt = contset(opt,'MaxNumPoints',100);
7      opt = contset(opt,'Singularities',1);
8
9      [x,v,s,h,f] = cont(@equilibrium,x0,[],opt);
10
11     bfpts = struct('gamma',gamma,'Gamma',Gamma,
12         'x',x,'v',v,'s',s);
13 end
```

Listing 4.1: Numerical continuation of the Greenland ice sheet model (Model 4.1).

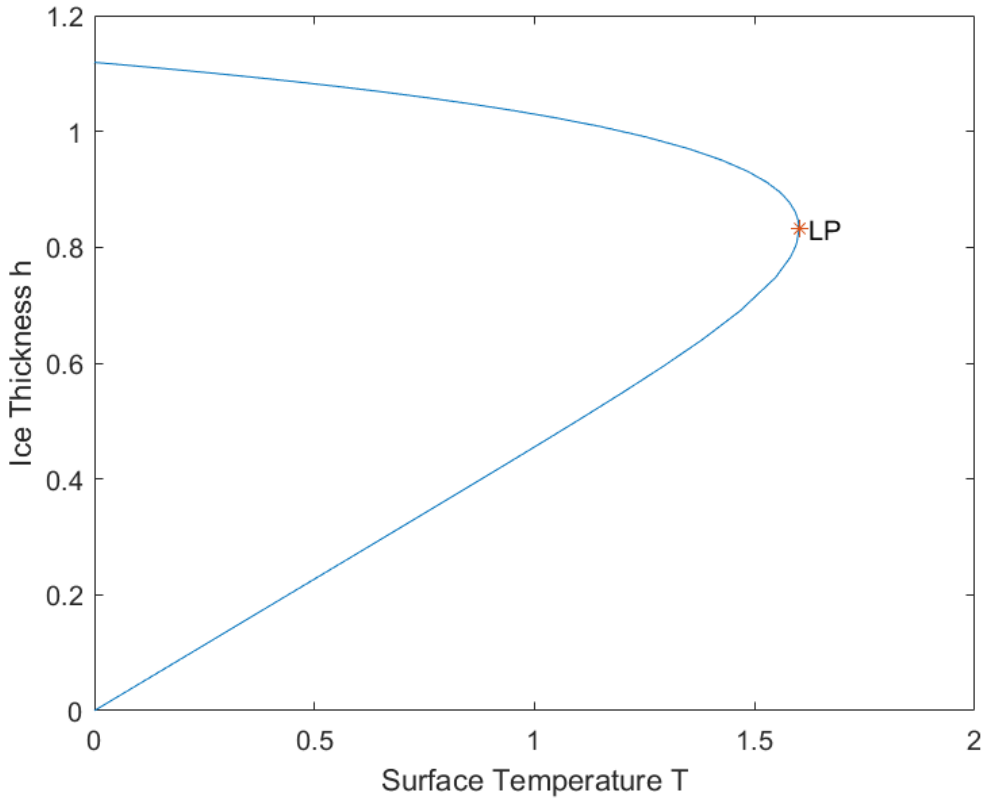


Figure 4.5: Bifurcation diagram of the Greenland ice sheet model (Model 4.1) for the parameters $\gamma = 440$ and $\Gamma = 0.005$ using `MATCONT`.

The bifurcation diagram for $\gamma = 440$ and $\Gamma = 0.005$ is depicted in Fig. 4.5. Compared to the previous plots (Fig. 4.1a and Fig. 4.2a), the result is without any issues: The code terminates after having found all points, it has no obstacles with initial points (here only one initial point is needed which was easy to calculate) and solutions in the neighborhood of the bifurcation point do exist.

This plot can be obtained by using the `MATCONT` file `cpl.m` (see Appendix A.1). The options for plotting regarding color and text position of LP were slightly modified by the author.

Benchmark Analysis Before executing the benchmark analysis, the maximum number of points was set to 500 to fairly compare the `MATCONT` performance with the other methods used for numerical continuation in Chapters 4.2.1 and 4.2.2. Note, that this means that the bifurcation diagram may consist of 500 points.

The median elapsed time for the code execution is 0.1548 seconds which is a significant improvement compared to the other two methods.

Limitations There are some minor disadvantages of `MATCONT`. The toolbox must be downloaded and consumes about 5.4 MB of disk space. Also, the user must get familiar with the package since at first, it is complicated to operate. [13, 60]

4.2.4 Comparison

As we have seen, the methods to numerically solve the Greenland ice sheet model (Model 4.1) produce different bifurcation diagrams.

The first two considered methods `vpasolve` and `fsolve` exhibited difficulties. The former displayed problems regarding the computation of points around the bifurcation point while the latter did not terminate after having computed the necessary bifurcation points and was highly dependent on the initial values.

The `MATCONT` toolbox, on the other hand, did show none of the aforementioned issues and, as a bonus, was significantly faster. To be exact, around 82 times faster than using `vpasolve` and around 44 times faster than `fsolve`.

To put it in a nutshell, the numerical continuation with `MATCONT` is preferable over the other methods. This is especially due to the more accurate computation of the bifurcation point and its visualization.

Therefore, in further analyses, the `MATLAB` function `bifurcation_points` created in Section 4.2.3 for using `MATCONT` will be used.

5. Sensitivity Analysis

In order to better capture processes in nature, their mathematical models are getting more and more complex. [7] An important aspect during model selection is performing a *sensitivity analysis*. With this, one can see how small variations in the input parameters influence the output of the model. [7, 36] From that, it may be possible to adjust the system or pay more attention to certain parameters. A parameter with a high influence on the model output is called *sensitive*. [36, Sec. 6.5] [11,64,73]

There are various methods for performing sensitivity analysis. These methods include the local and the global sensitivity analysis as well as the direct and the indirect method. [36, Sec. 6.5] [11,64,73,77] We focus in this thesis on the first two methods. After defining and explaining these, they are applied to the Greenland ice sheet model 4.1.

5.1 Local Sensitivity Analysis

To perform *local sensitivity analysis* on a model, each parameter is considered individually. This means that all parameters except for one are fixed. The parameter we are interested in is then varied slightly. [36, Sec. 6.5.1]

Let us consider the ODE as stated in Definition 3.1. The "change in the solution with respect to the parameter" is expressed through the partial derivatives. [36, p. 140] This brings us to the definition of local sensitivity.

Definition 5.1 (Local Sensitivity [36, Sec. 6.5] [11,64,73])**.** Local sensitivity *is the change of the solution of differential equations in relation to the parameter*

$$\mathcal{S}_\lambda(t) = \frac{dx(t,\lambda)}{d\lambda}. \tag{5.1}$$

More specifically in scalar notation,

$$\mathcal{S}_{i;\lambda}(t) = \frac{dx_i(t,\lambda)}{d\lambda}. \tag{5.2}$$

The sensitivity may also be called *dynamic sensitivity* because the model outputs are time-dependent trajectories. [64]

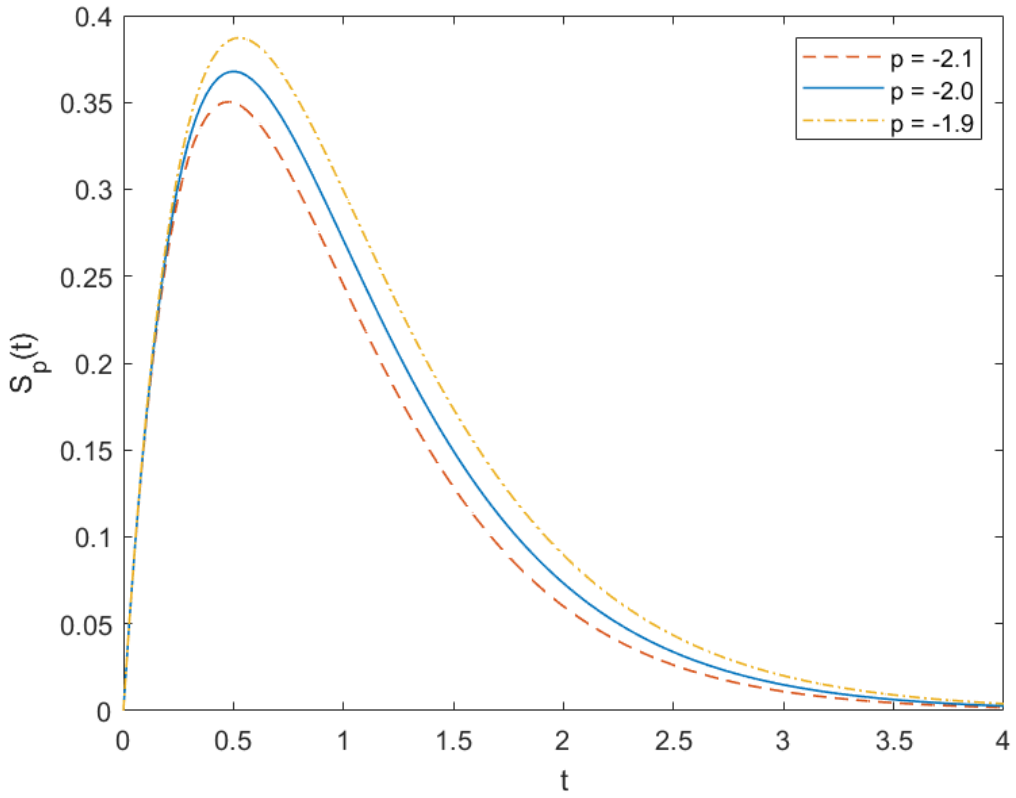


Figure 5.1: Local sensitivity analysis of the ordinary differential equation $x'(t) = px(t),\ x(0) = 2$ for varying parameter values $p \in \{-1.9, -2.0, -2.1\}$. This Figure is similar to [36, Fig. 6.5].

Example 5.1. Let us apply the definition onto a simple example. For this purpose, let $\mathcal{T} \in \mathbb{R}$, $\mathcal{X} \in \mathbb{R}$ and $\Lambda \in \mathbb{R}$ be open subsets. Consider an IVP as stated in Definition 3.2

$$x'(t) = f(t, x, p) = px(t), \quad x(0) = 2, \tag{5.3}$$

where $f : \mathcal{T} \times \mathcal{X} \times \Lambda \to \mathbb{R}$ and $p \in \mathbb{R}$. Further, $(t_0, x_0) \in \mathcal{T} \times \mathcal{X}$, where $t_0 = 0$, $x_0 = 2$. With the Existence and Uniqueness Theorem 1.2 from [8], there exists a unique solution $x(t) = 2\exp(pt)$ (see Sec. I.1.VII., separation of variables in [75]). Using Definition 5.1, the local sensitivity of the parameter p is

$$\mathcal{S}_p(t) = \frac{d}{dp} 2\exp(pt) = 2t\exp(pt). \tag{5.4}$$

Fig. 5.1 depicts the sensitivity $S_p(t)$ for different values of the parameter p. For the implementation, refer to Appendix A.2. We begin our analysis by considering only case $p = -2$ (blue solid line). The sensitivity curve has a maximum at $t \approx 0.5$. This peak is an indicator that the parameter is particularly sensitive at this point. As t progresses, the sensitivity approaches zero. This being said, the parameter p has a large influence on the result of the function around the peak value $t \approx 0.5$. Later this effect weakens.

Let us now compare the sensitivity curves with each other for small variations of the parameter p. The difference between the curves is not large, but for increasing values of p the peak becomes stronger and thus the parameter becomes more sensitive. [36, Def. 6.4]

All in all, Fig. 5.1 shows that the parameter p is locally sensitive. This is because as the parameter increases, so does the peak. [36, Sec. 6.5.1]

5.2 Global Sensitivity Analysis using Sobol Method

In this section, we look at *global sensitivity analysis* which is an extension of local sensitivity analysis. [36, Sec. 6.5.1] [11] The goal is to analyze all parameters at once, and also to see the interaction between the parameters in the whole input space. [3, 79] By performing such a global analysis, one can not only see how sensitive each parameter is, but also which parameter combinations affect the output and how the parameters of the model interact with each other. [23, 79]

There are various methods to perform global sensitivity analysis. Common techniques are: weighted average of local sensitivity analysis, partial rank correlation coefficient, multi-parametric sensitivity analysis, Fourier amplitude sensitivity analysis, and Sobol method. [79, Tab. 1]

We focus on the *Sobol method* because it is variance-based and thus one can determine to what degree an input parameter influences the variability of the output. [3] Further, in comparison to other mentioned techniques, the Sobol method has advantages in model independence, it can consider nonlinear relationships, and it is robust. The only negative aspect is that it is not computationally effective. [79]

In the following, the Sobol method is defined, explained and implemented.

5.2.1 Definition of the Sobol Method

The global sensitivity computed using the Sobol method is the ratio of the variance of the analyzed parameter or parameters to the variance of the entire model. Thus, the one-way, two-way, three-way etc. parameter interactions can be captured. [23,79]

The model we are interested in is defined through the function

$$f(x) = f(x_1, x_2, \ldots, x_n) \tag{5.5}$$

with $x = (x_1, x_2, \ldots, x_n)$ being the input parameters. We define the function f on the n-dimensional unit cube [23, 79]

$$K^n = \{x : \ 0 \leq x_i \leq 1, \ i = 1, \ldots, n\}. \tag{5.6}$$

Given that the mean $f_0 = \int_0^1 f(x)\,dx$ is a constant and "the integral of every summand over any of its own variables is zero" [3, p. 101], the function f can be decomposed into the sum [3, 23, 79]

$$f(x) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1, i\neq j}^{n} f_{ij}(x_i, x_j) + \ldots + f_{1\ldots n}(x_1, \ldots, x_n). \tag{5.7}$$

In the publications [65] and [79], this decomposition is also referred to as *analysis of variance (ANOVA) representation*.

Provided that f is integrable in K^n, the decomposition terms can also be written as [23, 79]

$$\begin{aligned} f_i(x_i) &= \int_0^1 \ldots \int_0^1 f(x) \prod_{k\neq i} dx_k - f_0, \\ f_{ij}(x_i, x_j) &= \int_0^1 \ldots \int_0^1 f(x) \prod_{k\neq i,j} dx_k - f_i(x_i) - f_j(x_j) - f_0. \end{aligned} \tag{5.8}$$

Note, that f_i is the term for one-way parameters, f_{ij} for two-way parameters, and so on. [23]

Now that the prerequisites have been introduced, we can give the definition for the partial and total variance since the Sobol method is variance-based as explained at the beginning of the chapter. [3]

Definition 5.2 (Partial Variance [3, 23])**.** *The* partial variance D_i *for the respective parameter* i *is defined as*

$$D_i = \int_0^1 f_i^2\,dx_i. \tag{5.9}$$

Definition 5.3 (Total Variance [3, 23])**.** D *is the* total variance

$$D = \int_{K^n} f^2(x)\,dx - f_0^2, \tag{5.10}$$

which can also be written as the sum of all partial variances

$$D = \sum_{i=D}^{n} D_i + \sum_{i=1, i\neq j}^{n} D_{ij} + \ldots + D_{1\ldots n}. \tag{5.11}$$

To determine the value of the partial variance D_{ij}, we need numerical computation. This is be discussed in Section 5.2.2.

The only thing remaining for this section is to define the Sobol sensitivity indices.

Definition 5.4 (Sobol Sensitivity Indices [79])**.** *The* Sobol sensitivity indices *for the i-th parameter are given by*

$$S_i = \frac{D_i}{D}, \tag{5.12}$$

and the Sobol sensitivity indices for the interaction of the i-th and j-th parameter is

$$S_{ij} = \frac{D_{ij}}{D}. \tag{5.13}$$

Proposition 5.1 (Property of Sobol Sensitivity Indices [23, 79])**.** *The sum of all Sobol sensitivity indices is* 1

$$1 = \sum_{i=1}^{n} S_i + \sum_{i=1, i\neq j}^{n} S_{ij} + \ldots + S_{1\ldots n}. \tag{5.14}$$

From the number n of parameters one can also determine the number of terms. Then, there are $2^n - 1$ parameter combinations. [23]

5.2.2 Numerical Computation

The numerical implementation of the Sobol method is based on the techniques described in "Sensitivity measures, ANOVA-like techniques and the use of bootstrap" by G. E. B. Archer, A. Saltelli and I. M. Sobol. [3]

The Sobol method can be performed with *Monte Carlo* integrals, a numerical method for simulation and integration. [3, 6]

Definition 5.5 (Monte Carlo Integration [6])**.** *Let x be a random variable that is uniformly distributed over $[0, 1]$. Further, consider a sequence $\{x_n\}$ that is sampled from the uniform distribution. In* Monte Carlo integration*, the one-dimensional unit interval $I(f) = \int_0^1 f(x)\,dx$ is approximated empirically by*

$$I_N(f) = \frac{1}{N} \sum_{n=1}^{N} f(x_n). \tag{5.15}$$

By the Strong Law of Large Numbers, $I_N(f)$ converges to $I(f)$ for $N \to \infty$.

The mean and variance of the model output f (introduced in Section 5.2.1) are approximated by Monte Carlo integrals as given in Definition 5.5. Therefore, for each parameter $x_i,\ i = 1, \ldots, n$, a *Sobol LP_T number sequence* consisting of N points is generated. This is a low-discrepancy sequence which consists of quasirandom numbers which converge fast and perform better than other sequences. [3, 23]

$$\begin{aligned} \hat{f}_0 &= \frac{1}{N} \sum_{m=1}^{N} f(x_m), \\ \hat{D} &= \frac{1}{N} \sum_{m=1}^{N} f^2(x_m) - \hat{f}_0^2. \end{aligned} \tag{5.16}$$

To numerically obtain the partial variances D_i, D_{ij}, D_{ijk}, and so on, the corresponding equations are also estimated via Monte Carlo integrals. The reformulation in [23, Eq. 22] shows the approximation for the partial variance D_i, which is $\hat{D}_i$:

Proposition 5.2 (Estimation of D_i [23])**.** *The partial variance D_i is approximated by Monte Carlo integrals and is given by*

$$\hat{D}_i = \frac{1}{N} \sum_{m=1}^{N} f\left(x_{-i,m}^{(1)}, x_{i,m}\right) f\left(x_{-i,m}^{(2)}, x_{im}\right) - \hat{f}_0^2. \tag{5.17}$$

Let us now make clear where the input parameters $x_{i,m}$, $x_{-i,m}^{(1)}$ and $x_{-i,m}^{(2)}$ come from and the reasoning behind this. For further explanation, we introduce the following notation [3]

$$x_{-i,m} = (x_{1,m}, \ldots, x_{i-1,m}, x_{i+1,m}, \ldots, x_{n,m}), \tag{5.18}$$

where $i = 1, \ldots, n$ and $m = 1, \ldots, N$.

Example 5.2. To demonstrate the notation given in Equation (5.18), let us look at an example: $x_{-1,m} = (x_{2,m}, x_{3,m}, x_{4,m}, \ldots, x_{n,m})$.

The publication of Archer, Saltelli and Sobol suggests to generate three Sobol LP_T number sequences, each of dimension $N \times n$. Then, $x_{i,m}$ is at the m-th stage of the Monte Carlo process and takes values from the i-th column of the first LP_T sequence. With Equation (5.18) and the explanation from Example 5.2, $x_{-i,m}^{(1)}$ takes values from all columns except the i-th from the second LP_T sequence. Similarly, $x_{-i,m}^{(2)}$ takes values from all columns except the i-th from the third sequence. [3]

Note that $x_{i,m}$ occurs twice in the input. The reason for this is enhancing the effect of the considered parameter and thus seeing how the result is affected by it. [3]

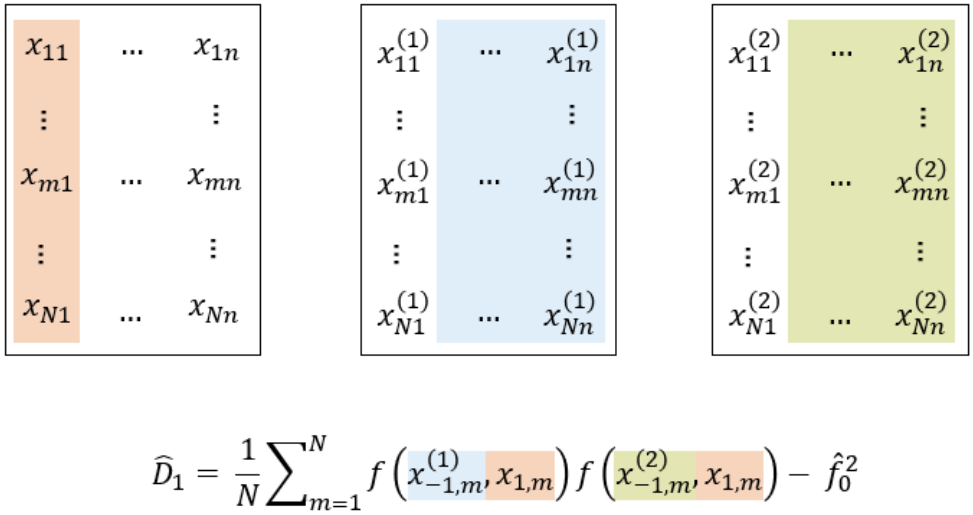


Figure 5.2: Illustration of the numerical computation of the Sobol method based on the explanations in [3].

An example for $\hat{D}_1$ is given by the illustration in Fig. 5.2. The matrices are the respective Sobol LP_T sequences of dimension $N \times n$. The orange highlighted area (left) is the considered parameter $x_{1,m}$, the blue area (middle) is $x^{(1)}_{-1,m}$, and lastly the green area (right) is $x^{(2)}_{-1,m}$. It is emphasized, that in the multiplication for each step, the considered parameter i stems from the same sequence (orange; left), while the other parameters each are from different sequences.

Analogously to the notation $x_{-i,m}$ from Equation (5.18), we introduce

$$x_{-i,j,m} = (x_{1,m}, \ldots, x_{i-1,\,m}, x_{i+1,\,m}, \ldots, x_{j-1,\,m}, x_{j+1,\,m}, \ldots, x_{n,m}), \tag{5.19}$$

where $i < j$ and $i = 1, \ldots, n$, $j = 1, \ldots, n$. We use this notation to approximate the partial variance D_{ij} of two parameters i and j. [3]

Proposition 5.3 (Estimation of D_{ij} [23])**.** *The approximated partial variance of the two-way parameter interaction D_{ij} is formulated as*

$$\hat{D}_{ij} = \frac{1}{N} \sum_{m=1}^{N} f\left(x^{(1)}_{-i,j,m}, x_{i,m}, x_{j,m}\right) f\left(x^{(2)}_{-i,j,m}, x_{i,m}, x_{j,m}\right) - \hat{D}_i - \hat{D}_j - \hat{f}_0^2. \tag{5.20}$$

Here, four inputs are needed. This is because two parameters, namely i and j, are considered and the effect of these two needs to be enhanced.

Finally, the equations for the numerically approximated Sobol sensitivity indices $\hat{S}_i$ can be derived using Definition 5.4.

Proposition 5.4 (Estimation of Sobol Sensitivity Indices [23])**.** *The Sobol sensitivity indices S_i can be approximated by Monte Carlo integrals as*

$$\hat{S}_i = \frac{\hat{D}_i}{\hat{D}}. \tag{5.21}$$

The Sobol sensitivity indices for the two-way, three-way, and so on, parameter interactions are estimated analogously.

Using the Sobol method and its numerical implementation has the advantage of being computationally efficient. [3] With this numerical computation at hand, we can now continue to implement this method in `MATLAB`.

5.2.3 `MATLAB` Implementation

The Sobol method performed with Monte Carlo integrals presented in Section 5.2.2 is implemented in `MATLAB`. Therefore, the function `sobol_method` is written in a way that the partial variances and Sobol indices of any model can be computed. The code can be found in Lst. 5.1 and is explained hereafter in further detail.

The function `sobol_method` takes the following input parameters: an integrable function `f`, the number of parameters `n`, the number of points `N`, and also three $N \times n$-dimensional matrices, each containing Sobol LP_T sequences as explained in Section 5.2.2 and Fig. 5.2. These matrices are referred to as `matrix_current` (orange/left, Fig. 5.2), `matrix_sample` (blue/middle, Fig. 5.2), and `matrix_resample` (green/right, Fig. 5.2).
The code outputs are the matrices `D` and `S` which each have the dimension $\binom{n}{\lceil n/2 \rceil} \times n$. Hereby, `D` contains the partial variances and `S` the corresponding Sobol indices. The structure of these matrices is explained shortly.

First the approximated mean $\hat{f}_0$ (`f0_hat`) and approximated variance $\hat{D}$ (`D_hat`) are computed according to Equation (5.16) (lines 2 and 3). Both $\hat{f}_0$ and $\hat{D}$ take their inputs from the i-th column in $x_{i,m}$, i.e. `matrix_current`.

In line 6, an empty matrix `D` for the partial variances $\hat{D}_i, \hat{D}_{ij}, \ldots$ is created. The entries are then filled in the `for`-loop, starting in line 8. Its structure is constructed in such a way that each column holds the indices for the one-way, two-way, etc. parameter interactions.

This means, for $n = 4$ parameters, the matrix `D` has the dimension $\binom{4}{2} \times 4$ and consists of the following indices:

1	12	123	1234
2	13	124	–
3	14	134	–
4	23	234	–
–	24	–	–
–	34	–	–

One can obtain the indices of each column by creating a vector `s` with the one-way, two-way, etc. parameter interactions using binomial coefficients (line 9). [47]

Another `for`-loop starting in line 12 is run to prepare for the input of the partial variance as introduced in Proposition 5.2. Therefore, two matrices `A` and `B` are constructed (lines 13 and 14) and filled (lines 17 to 24). Hence, the matrix `A` constitutes the input $(x^{(1)}_{-im}, x_{im})$, whereas `B` is $(x^{(2)}_{-i,m}, x_{i,m})$.

As visualized in Fig. 5.2, the values for the considered parameter are taken from the corresponding row (or rows) from `matrix_current`. We get $x^{(1)}_{-i,m}$ and $x^{(2)}_{-i,m}$ from the corresponding columns of `matrix_sample` and `matrix_resample`, respectively. The computation of the partial variances $\hat{D}_i, \hat{D}_{ij}, \ldots$ differs in the code for one-way parameter interactions or two- and more-way parameter interactions. The former uses `D_i = 0` (line 29, see Proposition 5.2) while the latter uses the sum of `D(a,i-1)` (line 27, see Proposition 5.3).

So, in the sense of the code, the respective columns of the last iteration are summed to be subtracted later as indicated in the formula. This means that the part $-(\hat{D}_i + \hat{D}_j)$ of the equation stated in Proposition 5.3 is subtracted due to the intuition described in T. Homma and A. Saltelli. [23]

The matrix `S` of Sobol indices can be determined using Proposition 5.4 which has the same structure as matrix `D`.

```
function [D, S] = sobol_method(n, N, fct,
    matrix_current, matrix_sample, matrix_resample)
 f0_hat = sum(fct(matrix_current))/N;
 D_hat = sum(fct(matrix_current).^2)/N - f0_hat^2;

 vec = 1:n;
 D = zeros(nchoosek(n,round(n/2)),n);

 for i = 1:n
  s = nchoosek(vec,i);
  k_size = size(s,1);

  for j = 1:k_size
   A = zeros(N,n);
   B = zeros(N,n);
   a = s(j,:);

   for k = 1:size(a,2)
    A(:,a) = matrix_current(:,a);
    B(:,a) = matrix_current(:,a);
   end

   b = setdiff(vec,a);
   A(:,b) = matrix_sample(:,b);
   B(:,b) = matrix_resample(:,b);

   if i ~= 1
    Di = sum(D(a,i-1));
   else
    Di = 0;
   end

   D(j,i) = sum(fct(A).*fct(B))/N - f0_hat^2 - Di;

  end
 end

 S = D/D_hat;
end
```

Listing 5.1: Implementation of the Sobol method with Monte Carlo integrals using MATLAB. It is based on the approach described in [3].

5.2.4 Validation of the Sobol Approach

The implementation `sobol_method` (see Lst. 5.1) is now verified with the analytical as well as computational results of the Ishigami and Homma model (Model 5.1) since these results are known from Section 3 in "Importance measures in global sensitivity analysis of nonlinear models" by T. Homma and A. Saltelli. [23]

Model 5.1 (Ishigami and Homma Model [27])**.** *The model introduced in T. Ishigami and T. Homma is a non-monotone and nonlinear function*

$$h(x_1, x_2, x_3) = \sin(x_1) + a\sin(x_2)^2 + bx_3^4 \sin(x_1), \tag{5.22}$$

with the variables a and b.

The three parameters x_1, x_2, x_3 have the uniform probability density function (PDF) [23, 27]

$$f_i(x_i) = \begin{cases} \dfrac{1}{2\pi}, & \text{for} \quad x_i \in [-\pi, \pi], \quad i = 1, 2, 3 \\ 0, & \text{else}, \end{cases} \tag{5.23}$$

The variables of Model 5.1 are selected as $a = 7$ and $b = 0.1$. [23, 27]

Before we can execute `sobol_method`, the input arguments of the code must be prepared. These steps are individual depending on the model and user preferences. The explanations for this example can be used as a basis.

The first step is to implement the Ishigami and Homma function h (see Lst. 5.2). Notice, the function parameters x_1, x_2 and x_3 are expressed as the columns of `x`. This later facilitates the computation of $f(x^{(1)}_{-im}, x_{im})$ and $f(x^{(1)}_{-im}, x_{im})$, respectively, since these inputs were constituted through the matrices `A` and `B`, respectively.

```
1 function y = ishigami_homma(x)
2     a = 7; b = 0.1;
3     y = sin(x(:,1))+a*sin(x(:,2)).^2+b*x(:,3).^4.*sin(x
          (:,1));
4 end
```

Listing 5.2: Implementation of the Ishigami and Homma function. [27]

It remains the step of creating the matrices `matrix_current`, `matrix_sample`, and `matrix_resample`. The code for this can be found in Lst. 5.3.

Therefore, a $3n$-dimensional Sobol LP_T number sequence `p` is generated using the `MATLAB` function `sobolset` (line 4). Further, it is recommended to skip the first entries since these are often zero and thus the sequence is unbalanced. [54] In line 5, we take the first N entries from the Sobol LP_T number sequence with `net`. [48] Hence, `x` forms the basis for the generation of the matrices just mentioned.

The Sobol LP_T sequence is a *low-discrepancy sequence*, which means that its deviation from a uniform distribution is very low. [10, 5.5] [63] For this reason, we

assume the Sobol LP_T sequence to be uniformly distributed in $[0,1]^n$. This eases the computation regarding the transformation of random variables.

With this being said, `matrix_current` is an $N \times n$-dimensional matrix with values between $-\pi$ and π (lines 7, 8 and 10). The same procedure is applicable to the matrices `matrix_sample` (line 11) and `matrix_resample` (line 12).

After generating the input arguments, we can now compute the partial variances as well as the Sobol indices for the Ishigami and Homma function h with `sobol_method` (line 14).

```
1  n = 3;
2  N = 2^11;
3
4  p = sobolset(3*n, 'skip', 1e3);
5  x = net(p,N);
6
7  lower_bounds = [-pi -pi -pi];
8  upper_bounds = [pi pi pi];
9
10 matrix_current = x(:,1:n).*[upper_bounds - lower_bounds
      ]+lower_bounds;
11 matrix_sample = x(:,(n+1):2*n).*[upper_bounds -
      lower_bounds]+lower_bounds;
12 matrix_resample = x(:,(2*n+1):(3*n)).*[upper_bounds -
      lower_bounds]+lower_bounds;
13
14 [D, S] = sobol_method(n, N, @ishigami_homma,
      matrix_current, matrix_sample, matrix_resample)
15
16 X = categorical({'x_1','x_2','x_3'});
17 X = reordercats(X,{'x_1','x_2','x_3'});
18 bar(X, [S(1,1), S(1,1)+S(1,2)+S(2,2)+S(3,1); ...
19         S(2,1), S(2,1)+S(1,2)+S(3,2)+S(3,1); ...
20         S(3,1), S(3,1)+S(2,2)+S(3,2)+S(3,1)]);
21 xlabel('Parameters');
22 ylabel('(Total) Sobol Sensitivity Indices');
23 legend('$$\hat{S}$$','$$\hat{S}_T$$','Interpreter','
      Latex');
24 ylim([0 1])
```

Listing 5.3: Preparation, execution and visualization of the Sobol method for the Ishigami and Homma model (Model 5.1).

The outputs from `sobol_method` are compared side by side with the analytical (exact) results [23, Eq. 50-57, Table 2, Table 4] and can be viewed in Table 5.1. Further, the numerical results are visualized in the form of a bar plot (see Fig. 5.3, lines 16-24), which will be discussed later.

Since there are $n = 3$ parameters, the number of parameter interactions is $2^3 - 1 = 7$. [23] Thus, the first column of Table 5.1 shows these one-way, two-way, and three-way parameter interactions. While D_i are the values for the analytically computed partial variances, $\hat{D}_i$ are the values that resulted from the `sobol_method`. Further, the analytical S_i and numerical $\hat{S}_i$ values of the Sobol indices are compared.

The last two columns contain the (analytical and numerical) values of the *total Sobol indices* S_{Ti} and $\hat{S}_{Ti}$. These are not direct outputs of the code and must be calculated separately.

Definition 5.6 (Total Sobol Indices [79])**.** *The* total Sobol indices *indicate how much influence a single parameter has in total, i.e.,*

$$S_{T_i} = S_i + \sum_{i=1, i \neq j}^{n} S_{ij} + \ldots + S_{1 \ldots i \ldots n}. \tag{5.24}$$

From Definition 5.6, it can be implied that the estimation of the total Sobol indices is given as $\hat{S}_{T_i} = \hat{S}_i + \sum_{i=1, i \neq j}^{n} \hat{S}_{ij} + \ldots + \hat{S}_{1 \ldots i \ldots n}$.

In Table 5.1, the analytical and numerical results obtained with the Sobol method are summarized. The analytical calculations come from T. Homma and A. Saltelli and the numerical results from the execution of function `sobol_method`, as described earlier in this section.

First of all, we would like to emphasize that the results of the analytical and numerical calculations are quite identical as the difference between the numerically and analytically calculated results is very small. Slight deviations are due to rounding errors.

Variables	D_i (exact)	$\hat{D}_i$ (approx.)	S_i (exact)	$\hat{S}_i$ (approx.)	S_{Ti} (exact)	$\hat{S}_{Ti}$ (approx.)
x_1	4.3459	4.3320	0.3138	0.3138	0.5574	0.5620
x_2	6.125	6.0733	0.4424	0.4400	0.4442	0.4523
x_3	0.0	0.0364	0.0	0.0026	0.2410	0.2549
x_1x_2	0.0	0.0383	0.0	0.0028	-/-	-/-
x_1x_3	3.3737	3.3507	0.2436	0.2428	-/-	-/-
x_2x_3	0.0	0.0952	0.0	0.0069	-/-	-/-
$x_1x_2x_3$	0.0	-0.1229	0.0	-0.0089	-/-	-/-

Table 5.1: Analytical and computational results of the partial variances, Sobol indices and total Sobol indices of the Ishigami and Homma model (Model 5.1). The exact results are from [23, Eq. 50-57, Table 2, Table 4] and the approximated values are results from the output of the `MATLAB` code `sobol_method`.

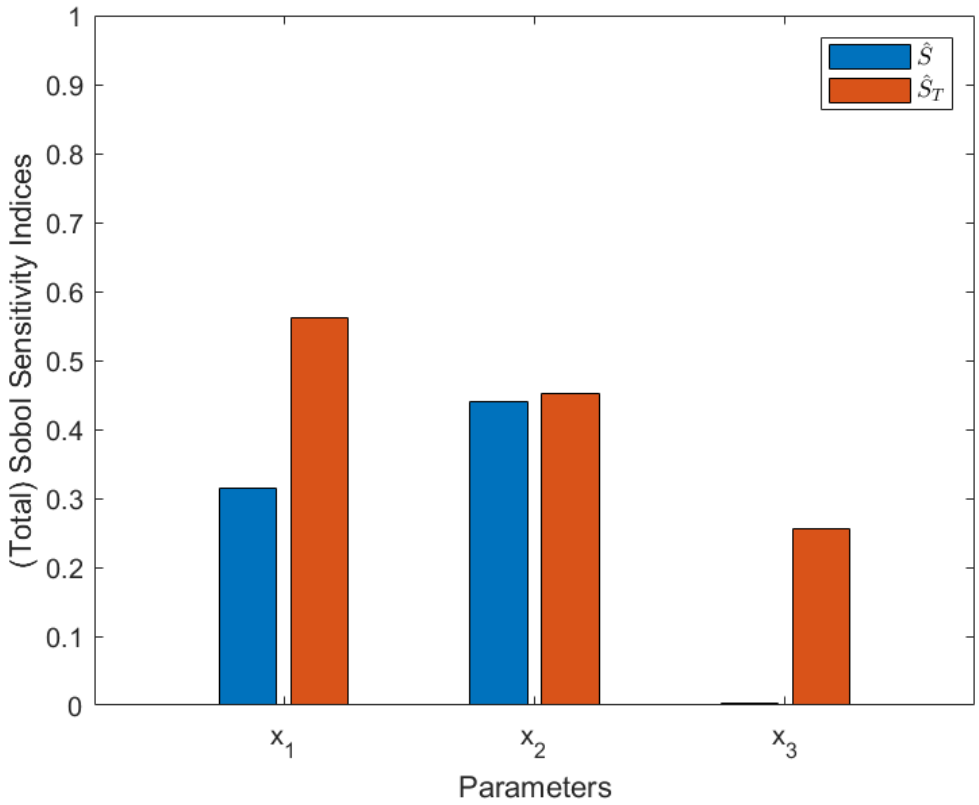


Figure 5.3: (Total) Sobol sensitivity indices of the Ishigami and Homma function. The visualization is based on [79, Fig. 3].

Next, let us look at the parameter interactions. They indicate that between the parameters x_1 and x_2, and also x_2 and x_3, is no interaction due to $S_{12} = 0$ and $S_{23} = 0$. Also, the three-way sensitivity indices S_{123} show no interaction. [79] That the partial variance D_3 is zero is an indicator that the parameter x_3 is negligible for Model 5.1. Nevertheless, it is not since there is a parameter interaction between x_1 and x_3. [23]

From Definition 5.6, we deduce that the total Sobol indices can only be calculated for the parameters x_1, x_2, and x_3. This can be seen from the empty cells in Table 5.1.

Ultimately, the results in Table 5.1 show that the `MATLAB` code `sobol_method` yields the correct results and can be used for further analyses.

Let us now plot the first-order sensitivity indices and the total sensitivity indices side by side. [79, Fig. 3]. The former is given by the blue bar (left, Fig. 5.3) and the latter by the orange bar (right, Fig. 5.3).

By looking at the orange bars (right), we observe that the parameter x_1 affects Model 5.1 by almost 60 %, x_2 by approximately 45 %, and the parameter x_3 affects the Model 5.1 by 25 %. The results also hint, that the parameter x_1 is the more critical since S_{x_1} has the highest value. [79]

Finally, we would like to emphasize once again that global sensitivity analysis using the Sobol method is from now on done using the implementation `sobol_method`.

5.3 Analysis of the Greenland Ice Sheet Model

To identify the sensitivity of the parameters and thus the quality of the model, we propose the following sensitivity analysis framework: First, we analyze the local sensitivity of the Greenland ice sheet model (Model 4.1). Second, we analyze the global sensitivity of the critical temperature. Lastly, we analyze the global sensitivity set of all tipping points. With this, we get a comprehensive overview of the behavior of the model in question and can evaluate its quality.

5.3.1 Local Sensitivity of the Greenland Ice Sheet Model

In this section, we want to determine the local sensitivity of the Greenland ice sheet model (Model 4.1) to analyze whether the parameters are well-suited and how the model output reacts to small changes of the parameters. The corresponding implementation can be found in Appendix A.2.

In agreement with Definition 5.1, the solution to this problem is required. However, this is difficult to solve analytically. Thus, the ODE is simplified by using the approximation approach:

$$h'(t) = -h^8(t) + \gamma\Gamma h(t) - T \approx \gamma\Gamma h(t) - T + \mathcal{O}(h^8). \tag{5.25}$$

Since the local sensitivity is the change of the solution in relation to the parameter (see Definition 5.1), approximating the ODE (Equation (5.25)) will not affect the result of the local sensitivity because the part $-h^8(t)$ does not contain the parameters γ, Γ, or T.

The approximated ODE (Equation (5.25)) is inhomogeneous. Its *general solution* can be found with the help of W. A. Adkins and M. G. Davidson. [1, p. 53] First, the homogenous ODE $h'_{\text{hom}}(t) = \gamma\Gamma h(t)$ is solved with separation of variables [75, Sec. I.1.VII]

$$h_{\text{hom}}(t) = c\exp(\gamma\Gamma t), \tag{5.26}$$

where c is a constant factor in $\mathbb{R}$. Next, the particular solution of $h'_{\text{p}}(t) = -T$ is solved according to [1, p. 53]

$$h_{\text{p}}(t) = \exp\left(\int \gamma\Gamma\, dt\right) \int \left((-T)\exp\left(-\int \gamma\Gamma\, dt\right)\right) dt = \frac{T}{\gamma\Gamma}. \tag{5.27}$$

Then, the general solution of the simplified ODE (Equation (5.25)) is the sum of the homogenous and particular solution, which is [1, p. 53]

$$h(t) = c \cdot \exp(\gamma\Gamma t) + \frac{T}{\gamma\Gamma}. \tag{5.28}$$

If we choose the initial condition to be $h(0) = 1 + \frac{T}{\gamma\Gamma}$, then it is $c = 1$ and the ODE (5.25) has the corresponding solution [75, p. 12]

$$h(t) = \exp(\gamma\Gamma t) + \frac{T}{\gamma\Gamma}. \tag{5.29}$$

Remark 5.1. Choosing a different initial condition leads to a different corresponding solution of the ODE. This also results in a different local sensititivity. In this thesis, we will not compare the sensitivity results for different initial conditions.

Let us now apply Definition 5.1 to each parameter of the solution 5.29. The local sensitivity of the melting sensitivity of ice γ is given by

$$S_\gamma(t) = \frac{1}{\Gamma t}\exp(\gamma\Gamma t) - \frac{T}{\gamma^2\Gamma}, \tag{5.30}$$

the local sensitivity of the atmospheric lapse rate Γ is

$$S_\Gamma(t) = \frac{1}{\gamma t}\exp(\gamma\Gamma t) - \frac{T}{\Gamma^2\gamma}, \tag{5.31}$$

and lastly, the local sensitivity of the surface temperature T is

$$S_T(t) = \frac{1}{\gamma\Gamma}. \tag{5.32}$$

Let us start with the analysis of the sensitivity of γ which is shown in the left graph in Fig. 5.4. Further, set Γ at 0.005 and T at 1.6.

As can be seen in the graph, the sensitivity curve rises faster after a certain point. This means, the γ parameter becomes more sensitive as time progresses. Thus, the melting sensitivity of the ice γ has a larger effect on the model after a certain point. Looking at small variations of the melting sensitivity γ, we find that the curves of $S_\gamma(t)$ differ slightly. However, the rapid increase remains the same.

The right graphs in Fig. 5.4 show the sensitivity of the atmospheric lapse rate Γ. The curves $S_\Gamma(t)$ are obtained by fixing $\gamma = 440$ and $T = 1.6$. Again, it can be seen that the parameter Γ becomes more sensitive over time. The parameter values of Γ are varied slightly around 0.005 to see whether small variations change the output. The curves are equally spaced and behave similarly.

Overall, the distance between the curves $S_\gamma(t)$ is close to zero until $T \approx 1.6$ and thereafter their distance is approximately equal as time progresses. Yet, for the sensitivity curves $S_\Gamma(t)$ on the left, the individual curves gradually increase.

Nonetheless, one can say that both the melting sensitivity of ice γ as well as the atmospheric lapse rate Γ have influence on the outcome. Therefore, the parameters of Model 4.1 are not negligible.

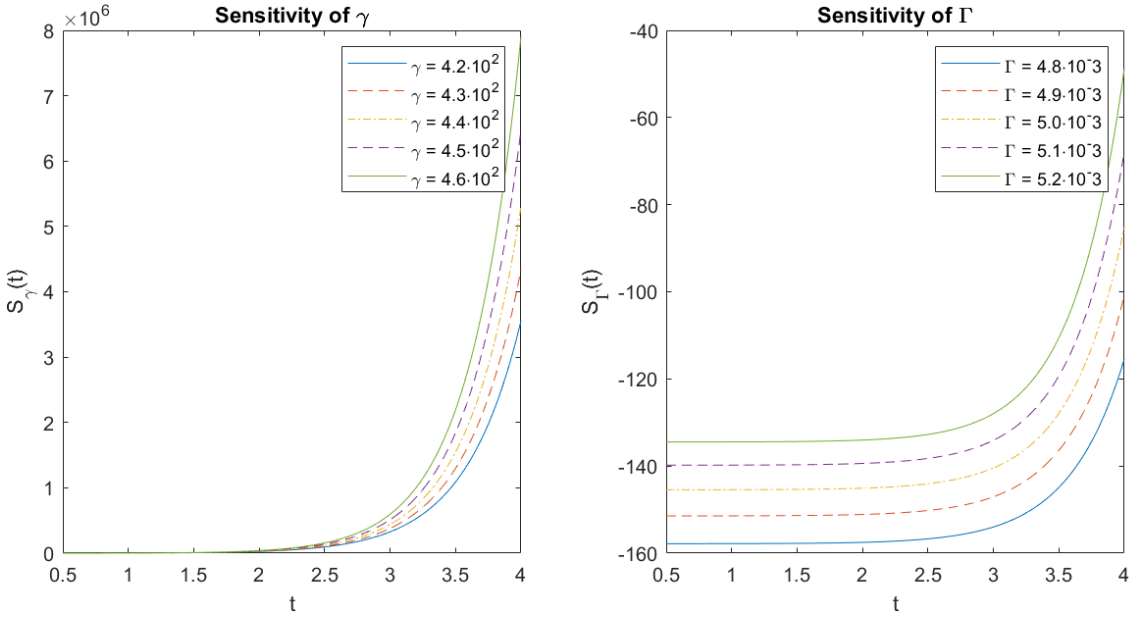


Figure 5.4: Local sensitivity analyses $S_\gamma(t)$ and $S_\Gamma(t)$ of the ice sheet model $h'(t) = \gamma\Gamma h(t) - T + \mathcal{O}(h^8)$, $h(0) = 1 + T/(\gamma\Gamma)$. Inspired by [36, Fig. 6.5].

Notice from Equation (5.32) that for fixed values of γ and Γ, the local sensitivity of the temperature $S_T(t)$ is a straight line. Thus, we cannot make any significant statements about the local sensitivity of the surface temperature.

5.3.2 Sobol Method applied to the Critical Temperature

In this section, we analyze the global sensitivity of the critical temperature T_c using the Sobol method. For this, the equation for the critical temperature is needed. To obtain it, we determine the extremum of the Model 4.1 as suggested in the publication of A. Levermann and R. Winkelmann. [34]

Proposition 5.5 (Equation of the Critical Temperature [34])**.** *The formula for the* critical temperature T_c *is given by*

$$T_c = -\left(\frac{\gamma\Gamma}{8}\right)^{8/7} + \gamma\Gamma\left(\frac{\gamma\Gamma}{8}\right)^{1/7} = 7\left(\frac{\gamma\Gamma}{8}\right)^{8/7}. \tag{5.33}$$

Proof. To get the equation for the critical temperature T_c, derive the right-hand side of Model 4.1 with respect to h

$$\frac{d}{dh}(-h^8 + \gamma\Gamma h - T) = -8h^7 + \gamma\Gamma. \tag{5.34}$$

Then, set this derivative (Equation (5.34)) equal to zero and solve for h. This results in the *critical surface elevation of the ice sheet* h_c

$$h_c = \left(\frac{\gamma\Gamma}{8}\right)^{1/7}. \tag{5.35}$$

Next, put the equation for h_c (Equation (5.35)) into the right-hand side of Model 4.1 and set this equation equal to zero. Solving for T gives the formula for the *critical temperature* T_c:

$$\begin{aligned} T_c &= -\left(\frac{\gamma\Gamma}{8}\right)^{8/7} + \gamma\Gamma\left(\frac{\gamma\Gamma}{8}\right)^{1/7} \\ &= \left(\frac{\gamma\Gamma}{8}\right)^{8/7}\left(-1 + \gamma\Gamma\frac{\left(\frac{\gamma\Gamma}{8}\right)^{1/7}}{\left(\frac{\gamma\Gamma}{8}\right)^{8/7}}\right) \\ &= \left(\frac{\gamma\Gamma}{8}\right)^{8/7}\left(-1 + \gamma\Gamma\left(\frac{\gamma\Gamma}{8}\right)^{-1}\right) \\ &= 7\left(\frac{\gamma\Gamma}{8}\right)^{8/7}. \end{aligned} \tag{5.36}$$

□

With the assumptions made in Section 5.2.1, the function f is then defined on the 2-dimensional unit cube as

$$f(x) = f(x_1, x_2) = 7\left(\frac{x_1 x_2}{8}\right)^{8/7}, \tag{5.37}$$

which can be implemented in `MATLAB` as follows:

```
1 function T = critical_temperature(x)
2     format longE
3     T = 7*((x(:,1).*x(:,2))/8).^(8/7);
4 end
```

Listing 5.4: `MATLAB` implementation of the function for the critical temperature.

Notice the number format style of the output is `longE` which is a "notation with 15 digits after the decimal point for double values". [39] We use that because the parameter Γ is in the range of 10^{-3}, which can later on lead to very small results. Hence, the format style is changed to still display these properly.

Variables	$\hat{D}_i$	$\hat{S}_i$	$\hat{S}_{Ti}$
γ	0.2324205677119697	0.5436742046962126	0.5828421151754957
Γ	0.1783348769887136	0.4171578848245044	0.4563257953037874
$\gamma\Gamma$	0.01674427057795258	0.03916791047928299	-/-

Table 5.2: Computational results of the partial variances, Sobol indices and total Sobol indices of the critical temperature. From the output of the `MATLAB` code `sobol_method`.

The `MATLAB` function `sobol_method` is executed as described in Section 5.2.4, i.e., by generating input arguments for the `MATLAB` code (see Appendix A.2). As for the PDF of the parameters, we assume that the parameters of the model are both uniformly distributed, i.e., $\gamma \sim U(240, 640)$ and $\Gamma \sim U(0.003, 0.007)$.

The results of the numerically computed partial variances $\hat{D}_i$, Sobol indices $\hat{S}_i$ and total Sobol indices $\hat{S}_{Ti}$ are gathered in the Table 5.2 and visualized in Fig. 5.5.

From the total Sobol indices $\hat{S}_{Ti}$ (see orange/right bar in Fig. 5.5 or fourth column in Table 5.2), one can obtain that the variable γ affects about 58 % of the model output whereas Γ influences the output around 46 %. Further, there is almost no interaction between both of the parameters ($\hat{S}_{\gamma\Gamma} \approx 0.0039$). Nevertheless, since both parameters of the ice sheet model have an influence, they are non negligible. [79]

The total Sobol indices indicate that both the melting sensitivity of ice γ as well as the atmospheric lapse rate Γ have comparable influence on the outcome. Since both parameters are a part of the linear factor of the Greenland ice sheet model (Model 4.1) and since they are being multiplied with each other, it is no surprise that both influence the outcome of the model to a similar degree.

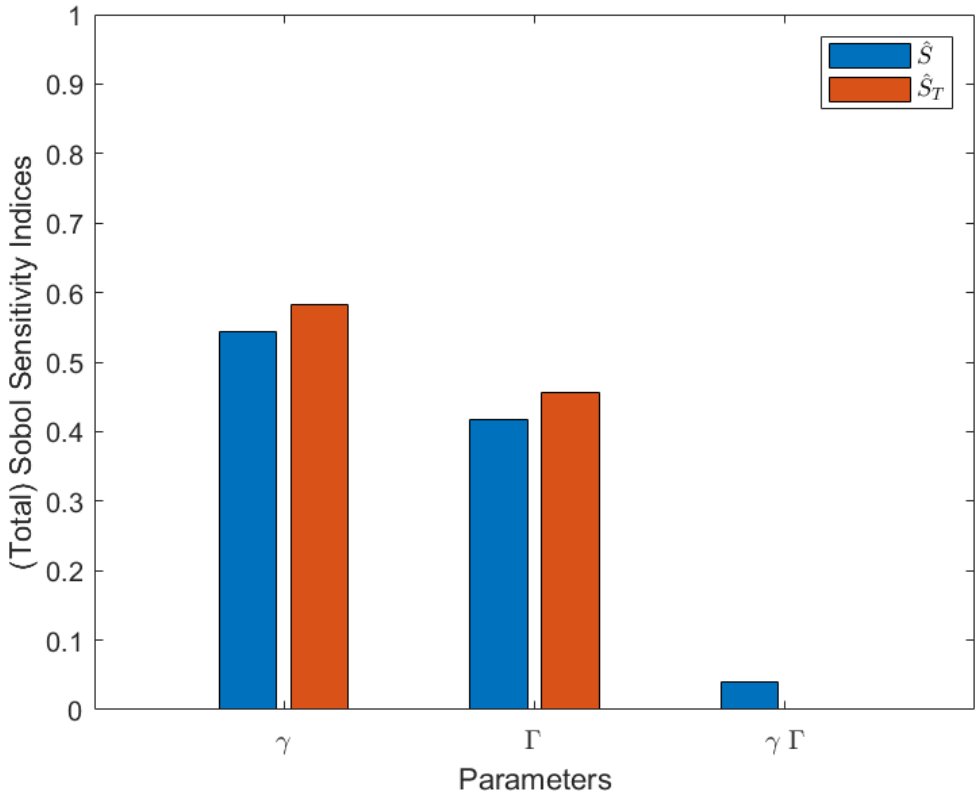


Figure 5.5: (Total) Sobol indices of the critical temperature T_c of the Greenland ice sheet model (Model 4.1). The visualization is based on [79, Fig. 3].

5.3.3 Sobol Method applied to the Set of Bifurcation Points

Let us here consider all bifurcation points of all bifurcation diagrams for the parameters $\gamma \sim U(240, 640)$ and $\Gamma \sim U(0.003, 0.007)$. In Fig. 5.6, each bifurcation curve (blue line) results from a parameter combination and the execution of Lst. 4.1. The orange point on the curve is the corresponding bifurcation point.

The bifurcation points of every bifurcation diagram constitutes a set of critical points. This brings up the question, how sensitive this set would be.

In implementing this idea, we proceed as follows: First, the set of critical points is interpolated resulting in a function. Subsequently, the Sobol indices of this function are computed.

Refer to Appendix A.2 for the implementation of the results presented in this section.

We begin by interpolating the elements in the set of critical points. Thus, we find a function $\hat{f}$ that has the critical temperature T_c (as derived in Proposition 5.5) as an input parameter and maps to the critical thickness h_c (see Equation (5.35)) of the Greenland ice sheet model (Model 4.1).

Given the visual form of the curve, a polynomial regression for the fifth degree is used. Thereby, the fifth degree forms such a polynomial that the discrete points are fitted as good as necessary. [76, Sec. 13.6]

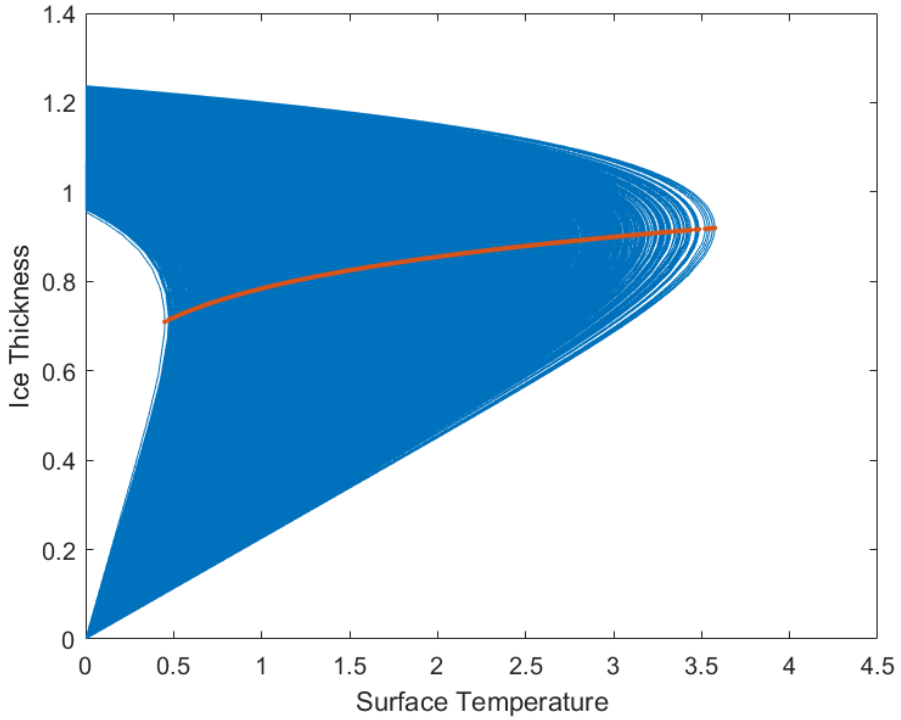


Figure 5.6: Bifurcation diagrams of the Greenland ice sheet model (Model 4.1) with their respective bifurcation points (orange points), results obtained with `MATCONT`.

This can be implemented in MATLAB using `polyfit` as shown: [49]

```
p = polyfit(Tc,hc,5);
```

The result obtained from `polyfit` is

$$\hat{f}(T_c) = 0.0013T_c^5 - 0.0156T_c^4 + 0.0753T_c^3 - 0.1920T_c^2 + 0.3127T_c + 0.6025, \quad (5.38)$$

which approximates the critical points of each bifurcation diagram. This function is implemented in MATLAB as follows

```
1 function y = critical_points(x)
2     format longE
3     T = 7*((x(:,1).*x(:,2))/8).^(8/7);
4     y = 0.0013*T.^5 -0.0156*T.^4 +0.0753*T.^3 -0.1920*T
          .^2 +0.3127*T +0.6025;
5 end
```

Listing 5.5: MATLAB implementation of the approximated function $\hat{f}(T_c)$.

The approximated function $\hat{f}(T_c)$ (orange solid line) is now plotted against the points from the set (blue circles) (see Fig. 5.7 on the left). We can see from the plot on the right (see Fig. 5.7) that the residuals are close to 0, which is an indicator for a good fit. [76, Sec. 13.1]

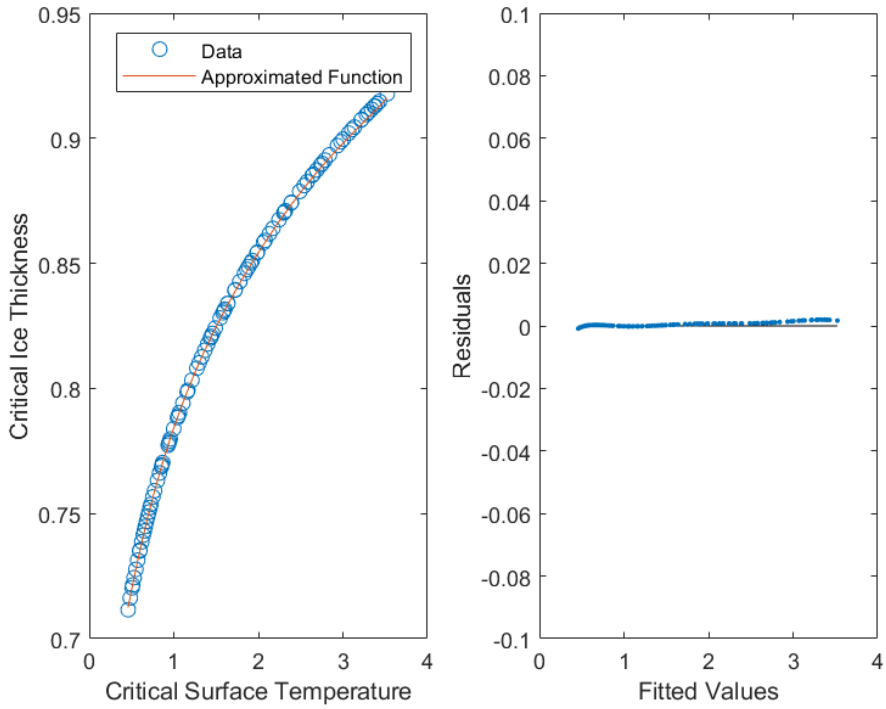


Figure 5.7: The plot on the left compares the result of the polynomial regression to the critical points. The plot on the right is a residual plot which demonstrates that the approximated function from the left plot is a good fit for the data.

Variables	$\hat{D}_i$	$\hat{S}_i$	$\hat{S}_{Ti}$
γ	0.001015955927761159	0.5610766835934421	0.5498436262423541
Γ	0.0008151096805689706	0.4501563737576459	0.4389233164065579
$\gamma\Gamma$	−0.00002033998477646293	−0.01123305735108801	-/-

Table 5.3: Computational results of the partial variances, Sobol indices and total Sobol indices of the set of bifurcation points, with the `MATLAB` code `sobol_method`.

For a global sensitivity analysis, the `MATLAB` function `sobol_method` (Section 5.2.4) is executed. The numerical results can be found in Table 5.3 as well as in Fig. 5.8.

The results are similar to the ones in Section 5.3.2. The melting sensitivity of ice γ has approximately 55 % influence on the outcome of the model whereas the atmospheric lapse rate Γ affects the outcome around 44 %.

This analysis puts the sensitivity analysis of the critical points in a different light. This is because now not only a single critical point is considered, but a function approximating all tipping points. We find that again, the results of the Sobol indices are very similar to the results of the Sobol indices from Section 5.3.2. The main difference between the two is the partial variance, which is mainly due to the fact that the inputs for the Sobol method are different.

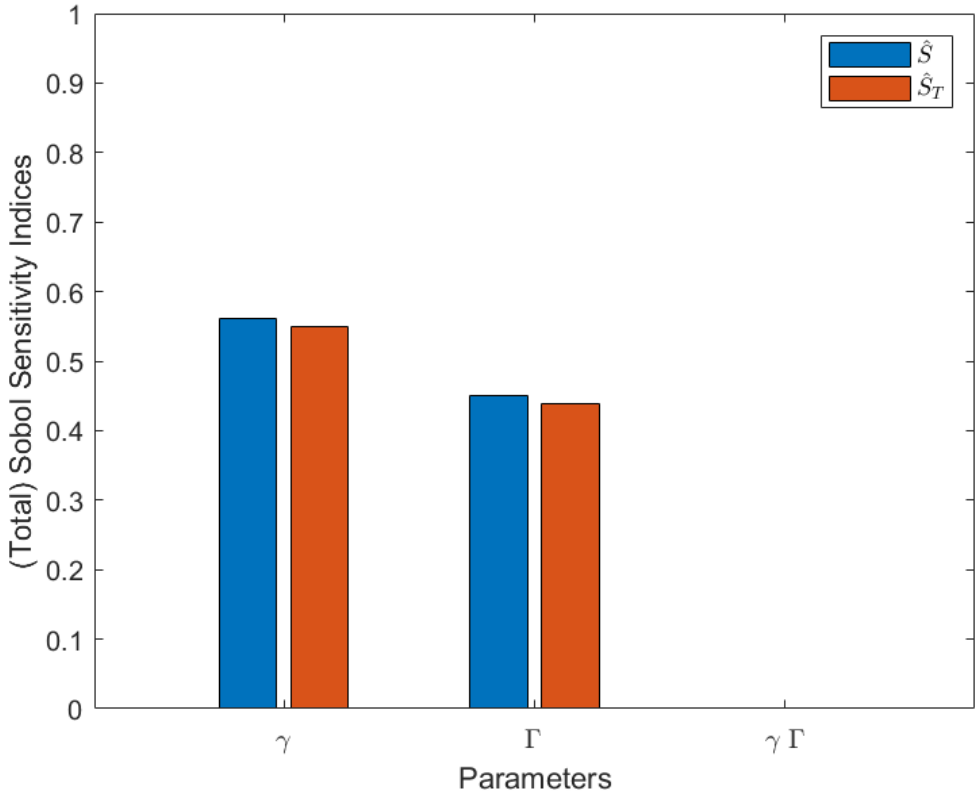


Figure 5.8: (Total) Sobol indices of the set of bifurcation points of the Greenland ice sheet model (Model 4.1). The visualization is based on [79, Fig. 3].

6. Link between Bifurcation and Sensitivity

We have already seen in Section 3.2 that varying the parameter of the ODE system can, for example, result in a change of stability of the equilibria. This means that the qualitative behavior of the system changes for small perturbations in the input parameters. [8, Sec. 8.1.3]

The characteristic of an input-output-connection is very similar to the sensitivity of an ODE. When analyzing a model, sensitivity analysis (see Chapter 5) can be combined with bifurcation analysis (as we have introduced in Section 3.2) to gain more understanding about the model and to enhance its quality. Hereby, the partial derivatives provide the connection of both analyses. [32, Sec. 2.3] [73]

Prior to this work, there is not yet enough information about the connection between sensitivities and bifurcations. Especially, performing only a sensitivity analysis leads to overlooked qualitative observation, whereas just bifurcation analysis leads to the fact that the sensitivity can be interpreted differently. [73]

In this chapter, we give two new approaches to analyze the sensitivity of the branches of equilibria of a bifurcation. In both approaches, the distance between a branch of equilibria and a *reference* branch of equilibria (that means branch of equilibria of an ODE with fixed parameters) is calculated and then the Sobol method is applied to these distances. The corresponding implementation is in the Appendix A.3.

The idea comes from visually looking at the bifurcation diagrams of the Greenland ice sheet model (Model 4.1) with the lowest and highest critical surface temperature (see blue lines in Fig. 6.1). Inserting the approximated function of all bifurcation points (Equation (5.38) derived in Section 5.3.3) into this visualization (see orange line in Fig. 6.1), we find that the bifurcation diagrams are stretched or compressed along this function.

For the realization, *Wasserstein-inspired* and *Wasserstein* distances are chosen for computing the distance between two branches of equilibria. The reason for this choice is that the Wasserstein distance is an optimal distance problem and thus suitable for our purposes. [74, Ch. 3]

The chapter is structured in a way that first the theory of Wasserstein distance is introduced. Subsequently, both approaches are explained by means of toy models. Finally, these approaches are compared and suggestions for improvement are given.

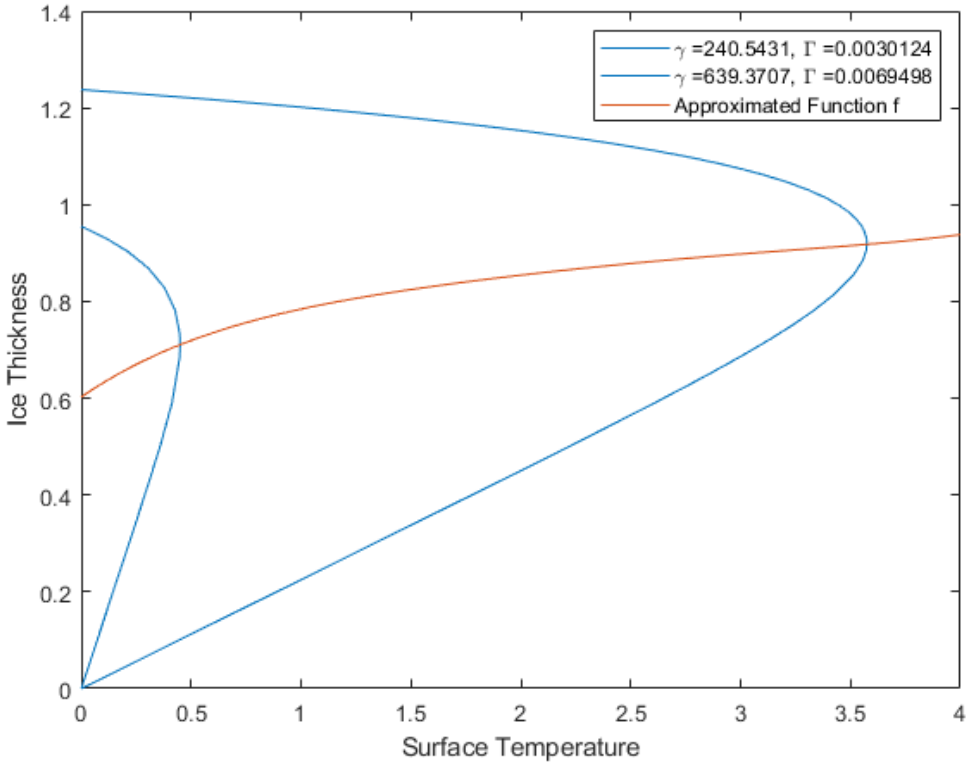


Figure 6.1: Bifurcation diagrams of the Greenland ice sheet model (Model 4.1) for the lowest and highest critical temperature with the interpolated function of all critical points.

6.1 Introduction to Wasserstein Distances

The *optimal transport problem*, also known as *Earth mover's distance*, addresses the problem of the "most efficient way of transforming one distribution of mass to another, relative to a given cost function". [30, p. 44]

G. Monge considered this problem as the minimal transport cost of moving a certain amount of soil from a source to another destination respecting the quantity in order to save costs. [74, Ch. 3] Thereby, the cost of transporting "one unit of mass along a certain distance was [determined as] the product of the mass by the distance". [74, p. 30]

This problem was then taken up in turn by L. Kantorovich, who proposed the optimal transportation cost for moving masses from different locations to multiple destinations, e.g., for bakeries distributing their goods to multiple cafés. [16, Sec. 1.1] This can also be represented as the optimal distance between probability measures, where both the quantity of goods from the bakeries and the goods consumed in the cafés are probability measures. [74, Ch. 3]

Such and other variants of distances are also called the *Kantorovich-Rubinstein distance*, or the *Wasserstein distance*. [74, Ch. 3] The Wasserstein distance for $p = 1$ is also known as the Earth mover's distance. [30]

This section will now give an introduction to p-Wasserstein distances. We begin with some auxiliary definitions:

Definition 6.1 (Separable Metric Space [28, Sec. 13.1])**.** *A metric space* (Ω, τ) *is* separable *if there exists a countable and dense subset of* Ω*.*

Definition 6.2 (Polish Space [28, Sec. 8.3])**.** *A* Polish space *is a separable topological space whose topology is generated by a complete metric.*

Some examples of Polish spaces are $\mathbb{R}^d, \mathbb{Z}^d, (C([0,1]), \|\cdot\|_\infty)$, where $d \in \mathbb{N}$. [28, Sec. 8.3]

Definition 6.3 (Coupling [28, Def. 17.53])**.** *Let be* $(\Omega_1, \mathcal{A}, \mu)$ *and* $(\Omega_2, \mathcal{B}, \nu)$ *two probability spaces. A* coupling *of the probability measures* μ *and* ν *is a probability measure on* $(\Omega_1 \times \Omega_2, \mathcal{A} \otimes \mathcal{B})$ *with* $\pi(\cdot \times \Omega_2) = \mu$ *and* $\pi(\Omega_1 \times \cdot) = \nu$*.*

With those in mind, we can now define the p-Wasserstein distance as follows.

Definition 6.4 (p-Wasserstein Distance [74, Def. 6.1], [16, Def. 1.4.3])**.** *Let* $(\mathcal{X}, d)$ *be a Polish metric space. The* p-Wasserstein distance *between the probability measures* μ *and* ν *on* $\mathcal{X}$ *is defined as*

$$W_p(\mu, \nu) = \left(\inf_{\Pi(\mu,\nu)} \int_{\mathcal{X}} d(x,y)^p \, d\pi(x,y) \right)^{1/p}, \tag{6.1}$$

where $p \in [1, \infty)$ *and* $\Pi(\mu, \nu)$ *is a set of couplings between* μ *and* ν*.*

The closed-form solution of the p-Wasserstein distance is a basis for the application in Section 6.2.

Proposition 6.1 (Closed-Form Solution for the p-Wasserstein Distance [30, Eq. 7])**.** *Let* F_μ *and* F_ν *be the cumulative distribution functions (CDFs) of the probability measures* μ *and* ν*, respectively. Assume that the inverses of the CDFs exist. Then,* F_μ^{-1} *and* F_ν^{-1} *denotes the inverses of the CDFs. The* closed-form solution *for the* p*-Wasserstein distance is given by*

$$W_p(\mu, \nu) = \left(\int_0^1 |F_\mu^{-1} - F_\nu^{-1}|^p \, dz \right)^{1/p} \tag{6.2}$$

6.2 Wasserstein-Inspired Distance and Sobol Sensitivity

The goal of this section is to determine the global sensitivity of branches of equilibria using the idea of the closed-form solution of the 1-Wasserstein distance from Proposition 6.1 and the Sobol method (see Section 5.2.1). Thereby, we will apply the ansatz on one-dimensional ODEs with one parameter each.

We begin by introducing a *Wasserstein-inspired* distance, that is based on Proposition 6.1.

Proposition 6.2 (1-Wasserstein-Inspired Distance)**.** *Consider an autonomous one-dimensional ODE $x' = f(x, \kappa)$ as stated in Theorem 3.1 with the parameter vector $\kappa = (a, \lambda)^\top \in \mathbb{R}^2$, where λ is the bifurcation parameter, satisfying the conditions*

1. *Let $F_\pm(\lambda)$ be the positive or negative branch of equilibria of the ODE. There exists the inverse of $F_\pm(a)$, which is $F_\pm^{-1}(z)$, provided each branch of equilibria is bijective.*

2. *Let $G_\pm(\lambda)$ be the positive or negative branch of equilibria for fixed parameter $\hat{a}$. Its inverse is given as $G_\pm^{-1}(z)$. The positive or negative branch of equilibria for fixed parameter will be called the* reference *positive or negative branch of equilibria.*

Then, the 1-Wasserstein-inspired distance *between a branch of equilibria $F_\pm(z)$ and a reference branch of equilibria $G_\pm(z)$ is given as*

$$\widetilde{W} = \left(\int_0^1 \left| F_\pm^{-1}(z) - G_\pm^{-1}(z) \right| \, dz \right). \tag{6.3}$$

This Proposition will be applied in the following sections on simple models.

6.2.1 Quadratic Influence

Let us apply the 1-Wasserstein-inspired distance to a simple one-dimensional ODE that has a quadratic, linear and constant term. The peculiarity of this model is that the quadratic term is amplified by a parameter. With this, we want to determine the *quadratic influence*.

The following model meets the characteristics described in Proposition 6.2. It is inspired from [68, Sec. 3.1].

Model 6.1. *Consider an autonomous and one-dimensional ODE as defined in Definition 3.3*

$$x' = ax^2 + x + \lambda, \tag{6.4}$$

where $a \in [1, 5]$ is a parameter and λ is the bifurcation parameter.

The solid lines in Fig. 6.2 are the bifurcation diagrams for the parameter values $a \in [1, 5]$. The light-blue dashed line is the *reference bifurcation diagram* for $\hat{a} = \frac{1}{2}$. Note that the choice of the reference bifurcation diagram is arbitrary. In this section, we will work with $\hat{a} = \frac{1}{2}$.

We observe that the bifurcation diagram for $a = 1$ (dark-blue solid line) is close to the reference bifurcation diagram. The larger the values for the parameter a are, the narrower the curves of the bifurcation diagrams get (observe the green bifurcation diagram ($a = 5$)). Not only do they become thinner, but they also share an equilibrium $(x_*, \lambda_*) = (0, 0)$.

Remark 6.1 (Visualization of Bifurcation Diagrams)**.** Each bifurcation diagram is visualized as explained in Section 4.2.3.

Hypothesis The parameter a of the quadratic term of Model 6.1 has a rather small influence on the model. We justify this hypothesis with the fact that the bifurcation diagrams only get narrower and do not move along the axes.

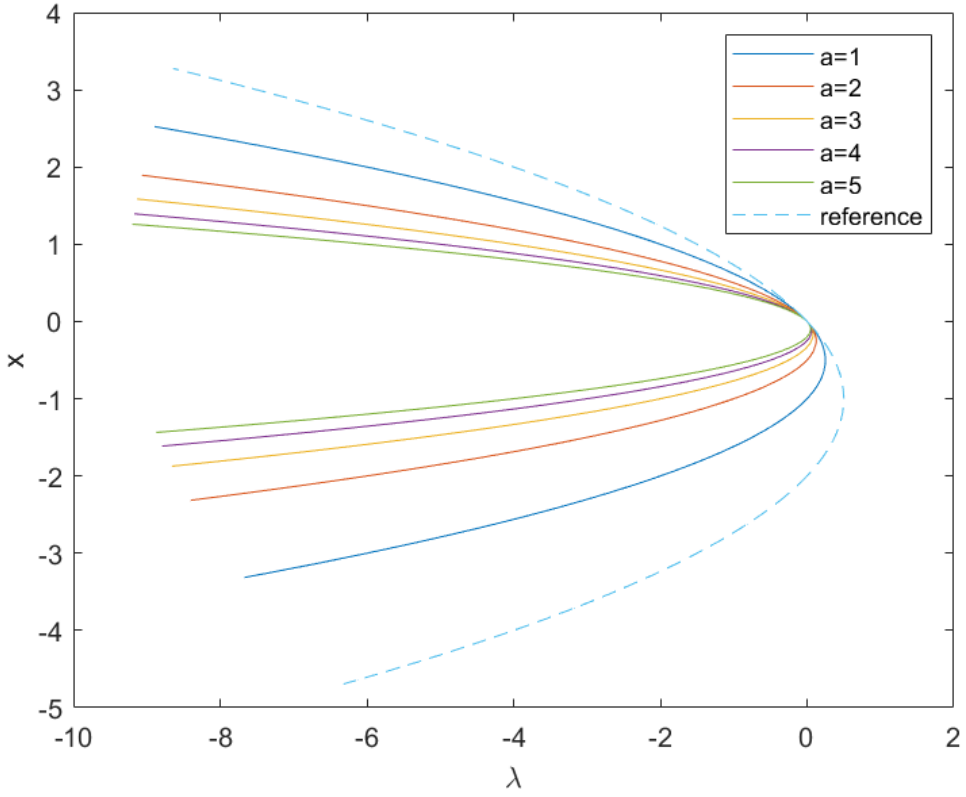


Figure 6.2: Bifurcation diagrams for different values of the parameter of Model 6.1. The light-blue dashed line is the reference bifurcation diagram for $x' = \frac{1}{2}x^2 + x + \lambda$.

To check if the hypothesis is met or not, we find the 1-Wasserstein-inspired distance $\widetilde{W}_q$ and then numerically compute the Sobol sensitivity from Section 5.2.3. The subscript q of the distance is just a notation to show that it is the 1-Wasserstein-inspired distance for the quadratic influence.

Note that the results are performed for the positive branch of equilibria. The calculations for the negative branch of equilibria are analogous, but are not shown here.

The positive branch of equilibria is calculated by solving the right-hand side of Model 6.1 for x.

$$F_+(\lambda) = \frac{-1 + \sqrt{1 - 4a\lambda}}{2a} \tag{6.5}$$

We assume $a \neq 0$. The inverse of the branch of equilibria $F_+(\lambda)$ is given by

$$F_+^{-1}(z) = \frac{1 - (2az + 1)^2}{4a} = -az^2 - z. \tag{6.6}$$

As already explained above, the choice for the reference branch of equilibria is arbitrary. In this example, we set $\hat{a} = \frac{1}{2}$. Then, the positive reference branch of equilibria is

$$G_+(\lambda) = -1 + \sqrt{1 - 2\lambda}, \tag{6.7}$$

and the inverse reference branch of equilibria is stated as

$$G_+^{-1}(z) = \frac{1 - (z + 1)^2}{2} = -\frac{1}{2}z^2 - z. \tag{6.8}$$

We can now apply Proposition 6.2 to Model 6.1.

$$\begin{aligned}
\widetilde{W}_q &= \int_0^1 |F_+^{-1}(a; z) - G_+^{-1}(\hat{a}; z)|\, dz \\
&= \int_0^1 |-az - z + \frac{1}{2}z^2 - z|\, dz = \int_0^1 |-az - 2z + \frac{1}{2}z^2|\, dz \\
&= \begin{cases} \displaystyle\int_0^1 |-az - z + \frac{1}{2}z^2 - z|\, dz = \frac{1}{3}\left(\frac{1}{2} - a\right) - 1 & \text{for } \; a < \frac{1}{2} \\ \displaystyle\int_0^1 |-\left(-az - z + \frac{1}{2}z^2 - z\right)|\, dz = -\frac{1}{3}\left(\frac{1}{2} - a\right) + 1 & \text{for } \; a > \frac{1}{2} \end{cases}
\end{aligned} \tag{6.9}$$

To obtain the case distinction, solve $-az - 2z + \frac{1}{2}z^2 = 0$ for z and consider $z > 0$. Also, $z < 1$ holds for any a.

The sensitivity of $\widetilde{W}_q$ is computed in Section 6.2.3.

6.2.2 Linear Influence

Another interesting approach is to analyze the Sobol sensitivity of the branches of equilibria of a simple one-dimensional ODE whose linear term is strengthened by a parameter.

The following model meets the characteristics described in Proposition 6.2 and is also inspired from [68, Sec. 3.1].

Model 6.2. *Consider an autonomous and one-dimensional ODE as stated in Definition 3.3*

$$x' = x^2 + bx + \lambda, \tag{6.10}$$

where $b \in [1, 5]$ is a parameter and λ is the bifurcation parameter.

The bifurcation diagrams of the branches of equilibria for different values of parameter b of Model 6.2 as well as the bifurcation diagram of the reference branches of equilibria for $\hat{b} = \frac{1}{2}$ are visualized in Fig. 6.3. The former are represented by solid lines, the latter by the light-blue dashed line.

As we can see, the leftmost dark-blue solid line is the bifurcation diagram for $b = 1$ and the rightmost green solid line is the bifurcation diagram for $b = 5$. All bifurcation diagrams have only one intersection point at $\lambda \approx 0$. The larger the parameter values b get, the more the bifurcation diagrams moves away from the reference bifurcation diagram. Note that there is only movement as the bifurcation diagrams do not get narrower or wider. They just move along both axes.

Hypothesis The parameter b of the linear term of Model 6.2 has significant influence on the model. Thereby, we suspect that the influence of the linear term bigger is than the influence of the quadratic term. This is due to the bigger movement of the bifurcation diagrams.

To check if the hypothesis is correct, we begin with the calculation of the 1-Wasserstein-inspired distance $\widetilde{W}_l$. Again, we only analyze the positive branch of equilibria.

The positive branch of equilibria is calculated in the same means as in Section 6.2.1.

$$K_+(\lambda) = \frac{-b + \sqrt{b^2 - 4\lambda}}{2} \tag{6.11}$$

The following equation is the inverse of the branch of equilibria $K_+(\lambda)$.

$$K_+^{-1}(z) = \frac{b^2 - (2z + b)^2}{4} = -z^2 - bz \tag{6.12}$$

Again, the choice of the reference branch of equilibria is arbitrary and is here $\hat{b} = \frac{1}{2}$. Then, the reference branch of equilibria is given by

$$L_+(\lambda) = \frac{-\frac{1}{2} + \sqrt{\frac{1}{4} - 4\lambda}}{2}, \tag{6.13}$$

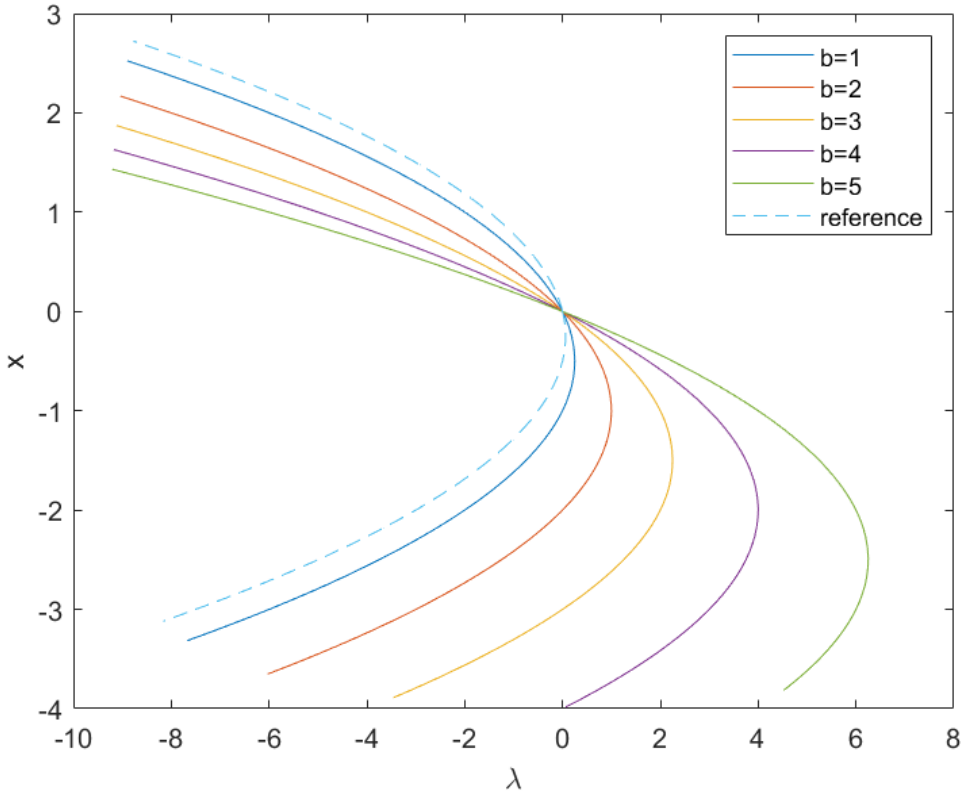


Figure 6.3: Bifurcation diagrams for different values of the parameter of Model 6.2. The light-blue dashed line is the reference bifurcation diagram, i.e., for $x' = x^2 + \frac{1}{2}x + \lambda$.

and the inverse reference branch of equilibria is

$$L_{\pm}^{-1}(z) = \frac{\frac{1}{4} - (2z + \frac{1}{2})^2}{4} = -z^2 - \frac{1}{2}z. \tag{6.14}$$

Let us apply Proposition 6.2 to Model 6.1.

$$\begin{aligned} \widetilde{W}_l &= \int_0^1 |K_+^{-1}(b;z) - L_+^{-1}(\hat{b},z)|\,dz \\ &= \int_0^1 |-z^2 - zb + z^2 + \frac{1}{2}z|\,dz = \int_0^1 |-zb + \frac{1}{2}z|\,dz \\ &= \begin{cases} \displaystyle\int_0^1 |-zb + \frac{1}{2}z|\,dz = -\frac{1}{2}b + \frac{1}{4} & \text{for } \; b < \frac{1}{2} \\ \displaystyle\int_0^1 |-\left(-zb + \frac{1}{2}z\right)|\,dz = \frac{1}{2}b - \frac{1}{4} & \text{for } \; b > \frac{1}{2} \end{cases} \end{aligned} \tag{6.15}$$

The case distinction is calculated by solving $-zb + \frac{1}{2}z$ for z and considering $z > 0$ as well as $z < 1$.

In Section 6.2.3, the sensitivity of $\widetilde{W}_l$ is computed.

6.2.3 Global Sensitivity Analysis of Wasserstein-Inspired Distance

In the last sections, the 1-Wasserstein-inspired distances $\widetilde{W}_q$ (see Equation (6.9)) and $\widetilde{W}_l$ (see Equation (6.15)) were calculated using Proposition 6.2. To conclude the analysis, the Sobol method will be applied to each of these distances.

The function `sobol_method` is executed as decribed in Section 5.2.3 and Section 5.2.4. As there is only one parameter (a or b) whose influence we want to measure, `n` is set to 1. Both of the parameters are in the range $[1, 5]$. Thus, the lower bound of the 3-dimensional Sobol LP_T number sequence is 1 and the upper bound is 5. As for the input function, it is simply the 1-Wasserstein-inspired distance $\widetilde{W}_q$ and $\widetilde{W}_l$, respectively.

Notice, that the functions $\widetilde{W}_q$ and $\widetilde{W}_l$, respectively, only consist of one parameter. Thus, the Sobol method computes the partial variance and Sobol index of the one-way parameter interaction. The Sobol index, however, is 1 in both cases due to Proposition 5.1. That is, since there is only one Sobol index, it is 1 by implication.

We have seen through the calculations and visualizations that the new approach of analyzing the global sensitivity of branches of equilibria of bifurcation diagrams is valid.

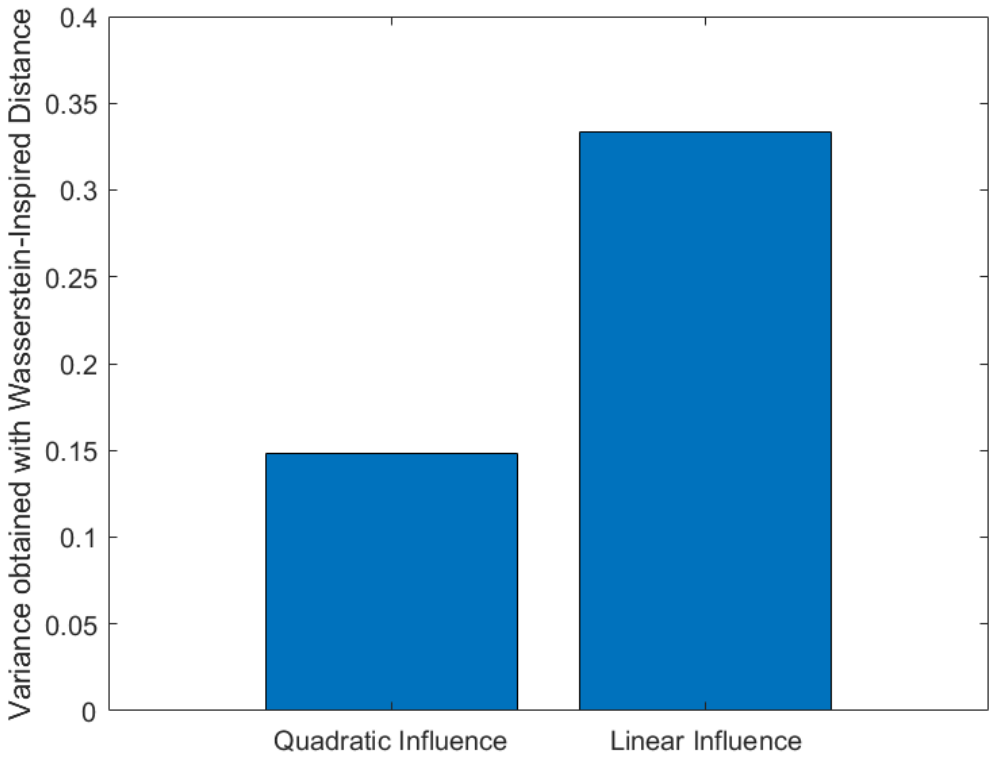


Figure 6.4: Results from the Sobol method applied to the 1-Wasserstein-inspired distance $\widetilde{W}_q$ (left bar) and $\widetilde{W}_l$ (right bar), respectively.

Fig. 6.4 shows the variance for $\widetilde{W_q}$ (obtained with Model 6.1; left bar) and $\widetilde{W_l}$ (obtained with Model 6.2; right bar), respectively. Observe that the variance of the quadratic influence is at 0.15 and less than the variance of the linear influence that is around 0.35.

Thus, the hypotheses made in Section 6.2.1 and Section 6.2.2 were confirmed.

However, there is a limitation. For the branches of equilibria, interpolation of the results from the numerical continuation (as described in Section 4.2.3) is needed. This will be shown in the next section.

6.3 Wasserstein Distance and Sobol Sensitivity

Here, we will apply the Definition 6.4 of the 1-Wasserstein distance to the simple models presented in Section 6.2.1 and Section 6.2.2. A new approach to the application of the Sobol method to Wasserstein distance is then presented.

The 1-Wasserstein distance is a distance between probability measures (see Definition 6.4). To be able to determine the distance between the branches of equilibria of different bifurcation diagrams, we represent these as *empirical distribution functions*.

Definition 6.5 (Empirical distribution function [28, Def. 5.22])**.** *Let $X_1, X_2, \dots$ be random variables. An* empirical distribution function *is defined as*

$$F_n : \mathbb{R} \to [0,1],\ x \mapsto \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{(-\infty,x)}(X_i). \tag{6.16}$$

Instead of analytically determining the empirical distribution function of the branches of equilibria of bifurcation diagrams, we utilize the `MATLAB` package `ws_distance` (`Version 1.0.1`). It "computes the 1- and 2-Wasserstein distances between two uniform probability distributions given through samples." [29]

With this package, we now determine the 1-Wasserstein distance W_q between the branches of equilibria of Model 6.1 and the reference branches of equilibria (Model 6.1 with $\hat{a} = \frac{1}{2}$). For Model 6.2, the 1-Wasserstein distance W_l is computed analogously.

We now proceed to the calculation of global sensitivity using the Sobol method with the `MATLAB` function `sobol_method`. The idea of the code explained in Section 5.2.3 remains the same, but the call of the input function changes.

The Sobol LP_T number sequence of dimension $N \times 1$ is created and has random values between 1 and 5. Hereby, the dimension and bounds are chosen in that way due to the considered model having only one parameter that is defined in $[1, 5]$ (see Section 6.2.1 and Section 6.2.2).
Instead of executing the function `f`, we now call the function `ws_distance` of the `MATLAB` package. Thereby, the inputs of the function are the bifurcation points of the positive or negative branch of equilibria of the *reference* bifurcation diagram

and the bifurcation points of the positive or negative branch of equilibria of *another* bifurcation diagram. Let us break down this statement:

Bifurcation points of the positive or negative branch of equilibria of the reference bifurcation diagram: These are the bifurcation points of the positive or negative branch of equilibria obtained with a `MATCONT` execution similar as in Section 4.2.3. These points are "fixed" in the code due to being the reference bifurcation diagram.

Bifurcation points of the positive or negative branch of equilibria of a bifurcation diagram: For these, we get the bifurcation points by executing the `MATCONT` function for a parameter input that comes from the Sobol LP_T number sequence.

Then, the 1-Wasserstein distance between these bifurcation points is computed with `ws_distance`. These steps are done for every row of the Sobol LP_T number sequence. Every result is then stored in a vector. With that, the partial variance and Sobol index is computed with Equation (5.16).

The result of the variance obtained with the 1-Wasserstein distance for Model 6.1 and Model 6.2 is constituted in Fig. 6.5. The variance for quadratic influence is around 0.035 and for the linear influence is around 0.045. There is a small difference between both variances. Nevertheless, the quadratic influence is lower than the linear influence.

6.4 Comparison and Outlook

Let us conclude this chapter with a comparison and outlook. We have presented two approaches for analyzing the global sensitivity of the branches of equilibria of bifurcation diagrams. Therefore, we looked into the 1-Wasserstein-inspired distance and also the 1-Wasserstein distance.

The statements were made on the basis of two simple models. In one we studied the sensitivity of the quadratic term parameter and in the other the sensitivity of the linear term parameter. The comparison of the two results can be seen in Fig. 6.6.

As can be seen, the variance for the quadratic influence is lower than the variance of the linear influence at each distance. This can also be seen from Fig. 6.2 and Fig. 6.3, where the behavior of the bifurcation diagrams for different parameters can be observed.

Notice that the scales for the variances of the two distances are quite different. This is due to different proceedings. The 1-Wasserstein-inspired distance, on one hand, computes the difference between branches of equilibria using their inverse (see Proposition 6.2). On the other hand, the 1-Wasserstein distance uses empirical distribution functions. As for the sensitivity analysis, the approach with the 1-Wasserstein-inspired distance uses the Sobol method directly as introduced in Section 5.2.3 while the procedure with the 1-Wasserstein distance makes modifications on the Sobol method.

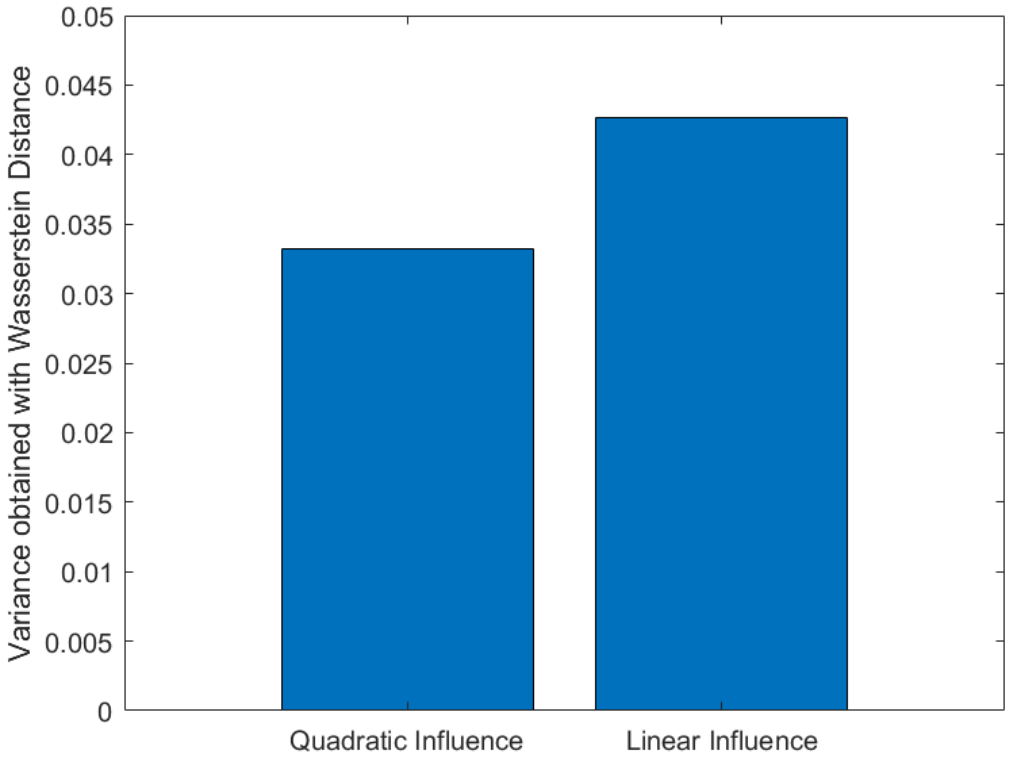


Figure 6.5: Results from the Sobol method applied to the 1-Wasserstein distance W_q (left bar) and W_l (right bar), respectively.

Although the scales are not equal, both variances calculated over the 1-Wasserstein-inspired distance and 1-Wasserstein distance have similar ratios, i.e., the quadratic influence is larger than the linear influence.

Further, it should be noted that the choice of the reference branches of equilibria can affect the results.

This can be explained by considering Fig. 6.3. Here, the reference branches of equilibria (light-blue dashed line) were selected such that they lie on the rightmost. The branches of equilibria for varying parameter move *away* from the light-blue dashed line. Hence, the distance gets bigger the bigger the parameter b gets.

Now imagine that the reference branches of equilibria would be in the position of the yellow solid line ($b = 3$), i.e., in the "middle" between all bifurcation diagrams. Then, the distance between the reference branches of equilibria that are on the right and on the left of said reference branches of equilibria is approximately the same. If we would create a distance vector, then it could look like this: $d = (1, 1, 2, 2, 3, 3, \ldots)^\top$. Hence, this kind of vector as an input for the partial variance and Sobol sensitivity may affect the outcome.

In the future, it would be interesting to see how much the choice of the reference branch of equilibria influences the sensitivity results.

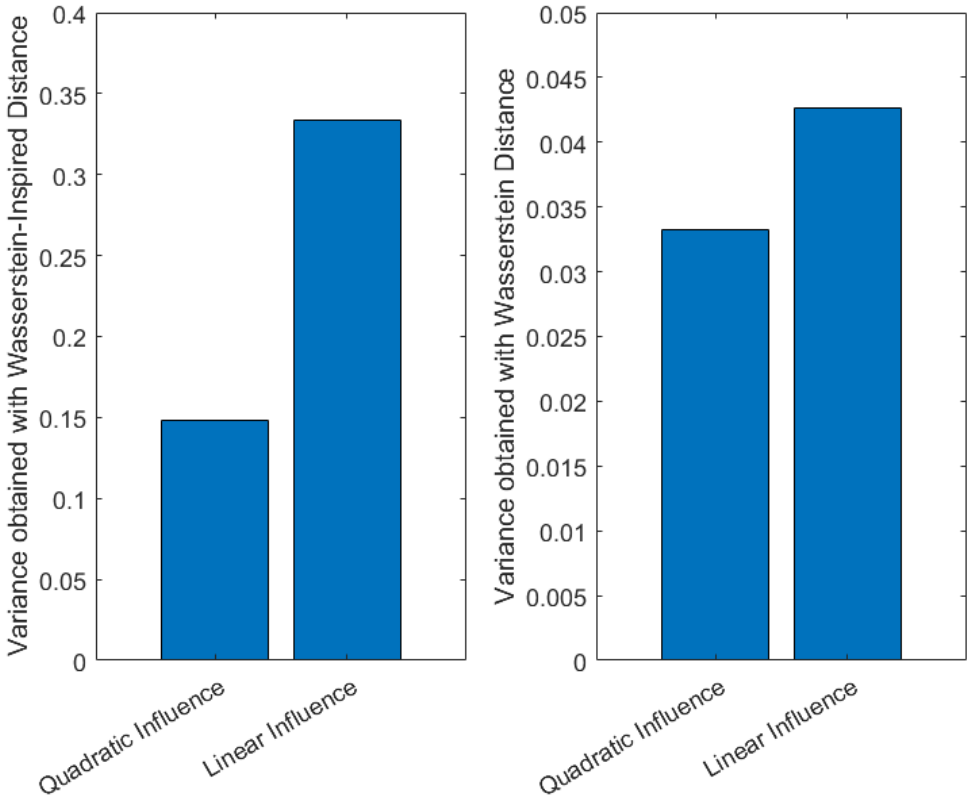


Figure 6.6: Comparison of the results gained through the 1-Wasserstein-inspired distance (left figure) and 1-Wasserstein distance (right figure).

Another point is that the code for the approach described in Section 6.3 could be extended to support a larger number of parameters and thus apply it to more complex models.

Still, the idea for analyzing the sensitivity of branches of equilibria of bifurcation diagrams using Wasserstein distances is interesting and offers clues to new insights. The findings from this chapter can be used as cornerstones.

7. Probabilistic Analysis of the Critical Temperature

Due to uncertainties in nature, one cannot find a specific fixed tipping point, in our case *the* critical temperature, for which the irreversible melt-off of the Greenland ice sheet is triggered. This means that these uncertainties are closely related the value of the critical temperature. [26]

The Greenland ice sheet model (Model 4.1) that is considered in this paper, consists inter alia of the parameters melting sensitivity of ice γ and atmospheric lapse rate Γ. The equation for the critical temperature T_c was derived and explained in Proposition 5.5. From this, we assume that the density of the parameters γ and Γ may affect the PDF of the critical temperature. Thus, by using a probabilistic approach, we want to determine the PDF of the critical temperature T_c to analyze what critical temperature is most likely.

First, we describe the theory of probabilistic analysis. The knowledge is then be applied to determine the PDF as well as the CDF of the critical surface temperature T_c. Last but not least, the analytical results are visualized using `MATLAB` and compared to numerical results.

7.1 Methodology

Fundamentals for the probabilistic analysis of the critical surface temperature are given.

Theorem 7.1 (Cumulative Distribution Function (CDF) [76, Def. 2.5, Thm. 2.8])**.** *A function $F_X : \mathbb{R} \to [0, 1]$ defined by*

$$F_X(x) = \mathbb{P}(X \leq x), \tag{7.1}$$

is called cumulative distribution function (CDF) *if and only if the following conditions are met:*

1. *F is non-decreasing: $x_1 < x_2 \Rightarrow F(x_1) \leq F(x_2)$*
2. *F is normalized: $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = 1$*
3. *F is right-continuous: $F(x) = F(x^+) = \lim_{y\to x, y>x} F(y) \quad \forall x$*

The proof of the conditions can be found in Theorem 2.8 of Wasserman. [76]

Definition 7.1 (Probability Density Function (PDF) [76, Def. 2.11])**.** *The* probability density function (PDF) *for a continuous random variable X is a function f_X that is defined as*

$$\mathbb{P}(a < X < b) = \int_a^b f_X(x)\,dx \quad \text{for } a \leq b. \tag{7.2}$$

Thereby, f_X satisfies $f_X \geq 0\ \forall x$ and $\int_{-\infty}^{\infty} f_X(x)\,dx = 1$.

Now that both the CDF and the PDF of a continuous random variable are defined, we can examine their relationship.

Proposition 7.1 (Relation between CDF and PDF [76, Def. 2.11])**.** *Given a CDF F_X as defined in Theorem 7.1 and a PDF f_X as defined in Definition 7.1, their relation is given by*

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\,dt, \tag{7.3}$$

and

$$F_X'(x) = f_X, \tag{7.4}$$

for F_X that is differentiable at all x.

Let there be a continuous random variable X with known PDF f_X. Consider a continuous random variable Y which is some transformation of X, i.e., $Y = g(X)$. With the following theorem, one can determine the PDF of Y. [22, Sec. 1.7.2]

Theorem 7.2 (Transformation of Continuous Random Variables [22, Thm. 1.7.1, p. 55])**.** *"Let X be a continuous random variable with PDF $f_X(x)$ and support $\mathcal{S}_X$. Let $Y = g(X)$, where $g(x)$ is a one-to-one differentiable function, on the support of X, $\mathcal{S}_X$. Denote the inverse of g by $x = g^{-1}(y)$ and let $\frac{dx}{dy} = \frac{d}{dy}(g^{-1}(y))$. Then the PDF of Y is given by*

$$f_Y(y) = f_X(g^{-1}(y)) \left|\frac{dx}{dy}\right|, \quad \text{for } y \in \mathcal{S}_Y, \tag{7.5}$$

where the support of Y is the set $\mathcal{S}_Y = \{y = g(x) :\ x \in \mathcal{S}_X\}$."

The proof of this theorem can be found in Theorem 1.7.1 of Hogg. It uses mainly Equation (7.1) given in Theorem 7.1 and then Proposition 7.1. [22, Thm. 1.7.1].

The *Mellin convolution* gives the PDF of the product $Z = XY$ of two non-negative and independent continuous random variables X and Y. [67, Sec. 4.2] It is defined as follows:

Definition 7.2 (Mellin Convolution [67, Eq. 4.2.4])**.** *Let X and Y be two non-negative and independent continuous random variables. The* Mellin convolution *of the product of the PDF $\varphi_1(x)$ of X and the PDF $\varphi_2(y)$ of Y with $x, y > 0$ is given as*

$$h(z) = \int_0^{\infty} \frac{1}{x} \varphi_2\left(\frac{z}{x}\right) \varphi_1(x)\,dx. \tag{7.6}$$

Assume that the space of the product of the two random variables X and Y is completely in the first quadrant. In that instance, the calculation of the Mellin convolution as defined in Definition 7.2 undergoes a distinction between cases. [17]

Proposition 7.2 (Case Differentiation for Mellin Convolution [17])**.** *The PDF of the product of independent random variables X and Y with the respective PDFs $\varphi_1(x)$ defined on (a,b), where $0 < a < b < \infty$ and $\varphi_2(y)$ defined on (c,d), where $0 < c < d < \infty$ is given by*

$$h(z) = \begin{cases} \displaystyle\int_a^{z/c} \frac{1}{x}\varphi_2\left(\frac{z}{x}\right)\varphi_1(x)\,dy & \text{for } ac < z < bc, \\ \displaystyle\int_a^{b} \frac{1}{x}\varphi_2\left(\frac{z}{x}\right)\varphi_1(x)\,dy & \text{for } bc < z < ad, \\ \displaystyle\int_{z/d}^{b} \frac{1}{x}\varphi_2\left(\frac{z}{x}\right)\varphi_1(x)\,dy & \text{for } ad < z < bd, \end{cases} \tag{7.7}$$

when $ad > bc$.

Case distinctions for Mellin convolution can also be made for $ad = bc$ and $ad < bc$. The respective formulas can be found in Chapter 2 of A. W. Glen, L. M. Leemis and J. H. Drew. The proofs can also be found in [17].

7.2 Analytical Computation

As seen from Proposition 5.5, the equation for the critical temperature is given by

$$T_c = 7\left(\frac{\gamma\Gamma}{8}\right)^{8/7}. \tag{7.8}$$

The goal of this section is to determine the PDF of the critical temperature which is yet unknown. However, in the publication of A. Levermann and R. Winkelmann, it was assumed that both the parameters γ as well as Γ are independent and uniformly distributed, such as [34]

$$\begin{aligned} \varphi_\gamma(x) &= \frac{1}{400}\mathbb{1}_{[240,640]}(x), \quad \text{and} \\ \varphi_\Gamma(x) &= \frac{1}{0.004}\mathbb{1}_{[0.003,0.007]}(x). \end{aligned} \tag{7.9}$$

Thus, we will use it as our advantage that the probability densities of the parameters of the critical temperature are known. With this being said, we first reformulate the equation for the CDF of T_c as done in Section 1.7.2 in R. V. Hogg, J. W. McKean and A. T. Craig. Let z be in the support of T_c, i.e. $z \in (0,\infty)$. [22]

$$F_{T_c}(z) = \mathbb{P}(T_c \le z) = \mathbb{P}\left(7\left(\frac{\gamma\Gamma}{8}\right)^{8/7} \le z\right) = \mathbb{P}\left(\gamma\Gamma \le 8\left(\frac{z}{7}\right)^{7/8}\right) = F_{\gamma\Gamma}\left(8\left(\frac{z}{7}\right)^{7/8}\right) \tag{7.10}$$

We continue with the last part of Equation (7.10). The random variable γ has support on the interval $(240, 640)$ and the random variable Γ has support on the interval $(0.003, 0.007)$. The space of the product of γ and Γ is therefore completely in the first quadrant. Thus, from the CDF of the product of the random variables γ and Γ follows the PDF using the Mellin convolution (see Definition 7.2 and Proposition 7.2). (*Step 1*)

After having determined the PDF $f_{\gamma\Gamma}$, we use the tranformation of continuous random variables (see Theorem 7.2), to get the PDF of the critical temperature T_c. (*Step 2*)

What is left to do, is to complete the calculation in Equation (7.10). Hence, by making use of the relation between CDF and PDF as shown in Proposition 7.1, we can find the CDF of the critical temperature. (*Step 3*)

The next subsections demonstrate the steps that were explained.

7.2.1 Step 1: Probability Density Function of $\gamma\Gamma$

Let us determine the PDF of the product of the random variables γ and Γ. To do this, we link to the last step of Equation (7.10), i.e.,

$$F_{\gamma\Gamma}\left(8\left(\frac{z}{7}\right)^{7/8}\right) = \int_0^{8(\frac{z}{7})^{7/8}} h_{\gamma\Gamma}(y)\,dy. \tag{7.11}$$

As mentioned earlier in Section 7.2, both random variables γ and Γ are independent and defined on positive intervals, respectively. Further, the product of these random variables lies completely in the first quadrant.

Hence, we can apply the Mellin convolution with case distinction (see Proposition 7.2). With the prerequisites introduced in said Proposition 7.2, it is $a = 0.003$, $b = 0.007$, $c = 240$, and $d = 640$. Hence, $ad > bc$.

Remark 7.1. Here, we choose φ_1 as the PDF of Γ, and φ_2 as the PDF of γ. However, the choice for φ_1 and φ_2 could also be the other way around, i.e., φ_1 as the PDF of γ, and φ_2 as the PDF of Γ. Then the condition $ad < bc$ holds and the first case from the Theorem in A. G. Glen, L. M. Leemis and J. H. Drew is applied. [17]

The conditions for the case differentiation for Mellin convolution are met (see Proposition 7.2) and the density function for the product of γ and Γ is calculated as follows:

$$\begin{aligned} h_{\gamma\Gamma}(y) &= \int_0^\infty \frac{5}{8}\frac{1}{x}\,dx \\ &= \begin{cases} \displaystyle\int_{0.003}^{y/240} \frac{5}{8}\frac{1}{x}\,dx = \frac{5}{8}\left(\ln(y) - \ln(0.72)\right) & \text{for } 0.72 < y < 1.68, \\ \displaystyle\int_{0.003}^{0.007} \frac{5}{8}\frac{1}{x}\,dx = \frac{5}{8}\ln\left(\frac{7}{3}\right) & \text{for } 1.68 < y < 1.92, \\ \displaystyle\int_{y/640}^{0.007} \frac{5}{8}\frac{1}{x}\,dx = \frac{5}{8}\left(-\ln(y) - \ln(4.48)\right) & \text{for } 1.92 < y < 4.48. \end{cases} \end{aligned} \tag{7.12}$$

To find the antiderivative of each section of the PDF, the *Fundamental Theorem of Calculus* is used. [31, Sec. 11.4]

Next, we go to the second step.

7.2.2 Step 2: Transformation of $\gamma\Gamma$ and Probability Density Function of T_c

In this step, we want to find the PDF of the random variable T_c using the transformation of continuous random variables. For this, let us check the conditions of Theorem 7.2.

The product of the independent and continous random variables γ and Γ is again a continuous random variable with the PDF given in Equation (7.12) and the support $\mathcal{S}_{\gamma\Gamma} = (0.72, 4.48)$.

Let be $g(y) = 7\left(\frac{y}{8}\right)^{8/7}$ the transformation function which is a one-on-one differentiable function. The inverse of g is given by $g^{-1}(z) = 8\left(\frac{z}{7}\right)^{7/8}$ and the derivative of the inverse with respective to z is $\frac{d}{dz}\left(g^{-1}(z)\right) = \left(\frac{z}{7}\right)^{-1/8}$.

Then, the PDF of the critical temperature T_c is given by

$$
\begin{aligned}
f_{T_c}(z) &= f_{\gamma\Gamma}\left(8\left(\frac{z}{7}\right)^{7/8}\right)\left|\left(\frac{z}{7}\right)^{-1/8}\right| \\
&= \begin{cases}
\frac{5}{8}\left(\ln(8\left(\frac{z}{7}\right)^{7/8}) - \ln(0.72)\right)\left(\frac{z}{7}\right)^{-1/8} & \text{for } 7\left(\frac{0.72}{8}\right)^{8/7} < z < 7\left(\frac{1.68}{8}\right)^{8/7}, \\
\frac{5}{8}\ln\left(\frac{7}{3}\right)\left(\frac{z}{7}\right)^{-1/8} & \text{for } 7\left(\frac{1.68}{8}\right)^{8/7} < z < 7\left(\frac{1.92}{8}\right)^{8/7}, \\
\frac{5}{8}\left(-\ln(8\left(\frac{z}{7}\right)^{7/8}) + \ln(4.48)\right)\left(\frac{z}{7}\right)^{-1/8} & \text{for } 7\left(\frac{1.92}{8}\right)^{8/7} < z < 7\left(\frac{4.48}{8}\right)^{8/7}.
\end{cases}
\end{aligned}
\tag{7.13}
$$

Note that the definition ranges for z for each section of the PDF given in 7.13 are also transformed with the transformation function $g(z) = 7\left(\frac{z}{8}\right)^{8/7}$.

We move on to the next and final step.

7.2.3 Step 3: Cumulative Distribution Function of T_c

Lastly, the CDF of the critical temperature T_c is determined using the relation between CDFs and PDFs (see Proposition 7.1). For this purpose, the antiderivative is found for each section of the PDF calculated in Section 7.2.2 (see Equation (7.13)). The calculations in this part of thesis are performed with `Mathematica` (`Version 13.1.0.0`).

Then, the CDF of the critical temperature T_c is given by

$$
\begin{aligned}
F_{T_c}(z) &= \int f_{T_c}(z)\,dz \\
&= \begin{cases}
z^{7/8}\left(-0.26849 + 0.797108\ln(z)\right) + c_1 & \text{for } 7\left(\frac{0.72}{8}\right)^{8/7} < z < 7\left(\frac{1.68}{8}\right)^{8/7}, \\
0.771872\,z^{7/8} + c_2 & \text{for } 7\left(\frac{1.68}{8}\right)^{8/7} < z < 7\left(\frac{1.92}{8}\right)^{8/7}, \\
z^{7/8}\left(1.93388 - 0.797108\ln(z)\right) + c_3 & \text{for } 7\left(\frac{1.92}{8}\right)^{8/7} < z < 7\left(\frac{4.48}{8}\right)^{8/7}.
\end{cases}
\end{aligned}
\tag{7.14}
$$

As the PDF (see Equation (7.13)) is continuous, the CDF (see Equation (7.14)) has to be continuous as well. With this fact and the properties of CDFs (see Theorem 7.1), the integration constants c_1, c_2 and c_3 can be determined as the following calculation shows.

$$
\begin{aligned}
F_{T_c;1}\left(7\left(\frac{0.72}{8}\right)^{8/7}\right) &\stackrel{!}{=} 0 && \Rightarrow c_1 \approx 0.45 \\
F_{T_c;1}\left(7\left(\frac{1.68}{8}\right)^{8/7}\right) &\stackrel{!}{=} F_{T_c;2}\left(7\left(\frac{1.68}{8}\right)^{8/7}\right) && \Rightarrow c_2 \approx -0.6 \\
F_{T_c;2}\left(7\left(\frac{1.92}{8}\right)^{8/7}\right) &\stackrel{!}{=} F_{T_c;3}\left(7\left(\frac{1.92}{8}\right)^{8/7}\right) && \Rightarrow c_3 \approx -1.8 \\
F_{T_c;3}\left(7\left(\frac{4.48}{8}\right)^{8/7}\right) &\stackrel{!}{=} 1 && \Rightarrow \text{that is true}
\end{aligned}
\tag{7.15}
$$

Consequently, we can proceed with the visualization of the results and the comparison with numerical results.

7.3 Visualization and Interpretation

As important the analytical computation of the PDF and CDF of the critical temperature T_c is, the numerical simulation of the PDF and CDF is also significant. By doing this, we can validate the results and gain more information about them.

The code used during this chapter can be found in Appendix A.4.

7.3.1 Visualization of the Probability Density Function

The analytical computation of the PDF of the critical surface temperature T_c was shown in Section 7.2.2. We visualize the piecewise densities as indicated in Fig. 7.1. The blue line depicts $f_{T_c;1}$, the orange line describes $f_{T_c;2}$, and the yellow line shows $f_{T_c;3}$ (for all pieces of the PDF, see Equation (7.13)).

The histograms in Fig. 7.1 are constructed from empirical data. The black solid bars are obtained from the empirical data from `MATCONT`. The purple dotted bars show the histogram that is obtained using the analytical formula of the critical surface temperature (see Proposition 5.5).

We can visually see that all graphs coincide. The mathematical comparison and interpretation are done in Section 7.3.3. The following paragraphs explain the process of simulating such data.

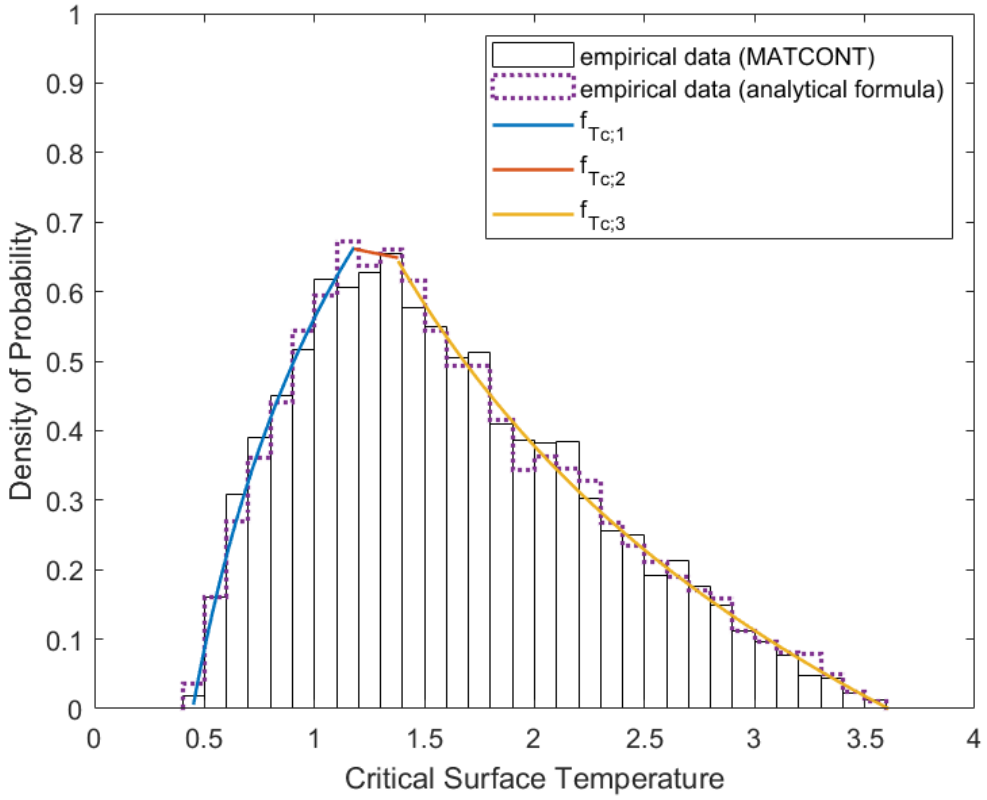


Figure 7.1: Probability density function of the critical surface temperature. The analytical result is depicted by the blue ($f_{T_c;1}$), orange ($f_{T_c;2}$) and yellow ($f_{T_c;3}$) lines. Both histograms obtained from empirical data show the normalized frequency of the critical temperature.

Empirical Data from `MATCONT` The function `bifurcation_points` was introduced in Section 5.2.3. For Model 4.1, the function numerically computes the bifurcation points and marks the critical point for input values γ and Γ. Remember, that these are uniformly and independent distributed as described in Section 7.2. The goal is to numerically simulate values for the critical surface temperature T_c, hence we make use of the the function `bifurcation_points` and execute this $n = 5000$ times to generate critical points T_c.

First, the parameter distributions are specified using `makedist` and then within a `for`-loop, the function `bifurcation_points` is being executed. The `MATLAB` function `makedist` generates a probability distribution object. [43] Thereby, a random number is drawn using `random`. [50] These n results are saved into a `.mat` file using `save`, so they can be visualized in a separate step without having to compute all the results again. [51, 52]

The `MATLAB` function `histogram` is then being used to plot a histogram of the critical points from each iteration step. The bin width is set to 0.1. [41]

Empirical Data from Analytical Formula There is another way of numerically simulating the critical surface temperature T_c. For this, we take the analytical formula for the critical temperature (see Equation (5.33) derived in Proposition 5.5), which we also use for the analytical calculation of the PDF. Then we simulate the uniformly distributed random numbers for γ and Γ and insert them into the analytical formula for the critical temperature.

Remark 7.2 (Note about Visualization)**.** Even though it takes a couple of minutes to generate one `.mat` file, it saves time for later computations, like visualizations and trials for different ideas. To access the results of the numerical continuation, use the function `load`. [42]

7.3.2 Visualization of the Cumulative Distribution Function

Let us consider the CDF of the critical surface temperature T_c. The piecewise functions that were computed in Section 7.2.3 are visualized in Fig. 7.2. Also here, for Equation (7.14), the blue line depicts $F_{T_c;1}$, the orange line describes $F_{T_c;2}$, and the yellow line shows $F_{T_c;3}$.

As for the numerical validation, an empirical CDF (black dashed line) of the critical points T_c is plotted using the `MATLAB` function `cdfplot`. [37] These points were generated with `MATCONT` as in Section 7.3.1. The purple dotted line is also an empirical CDF of the critical points T_c which were obtained with the analytical formula (see Proposition 5.5 and Section 7.3.1).

The range of the data is approximately between 0.4 and 3.6, so the analytical results are in agreement with the numerical results. The curve is steep from approximately 1 to 2 and it flattens at around 2.8.

About 30 % of the data have a critical surface temperature less than 1.2, about 50 % of the data have a critical surface temperature less than 1.5, and 80 % of the data have a critical surface temperature less than 2.

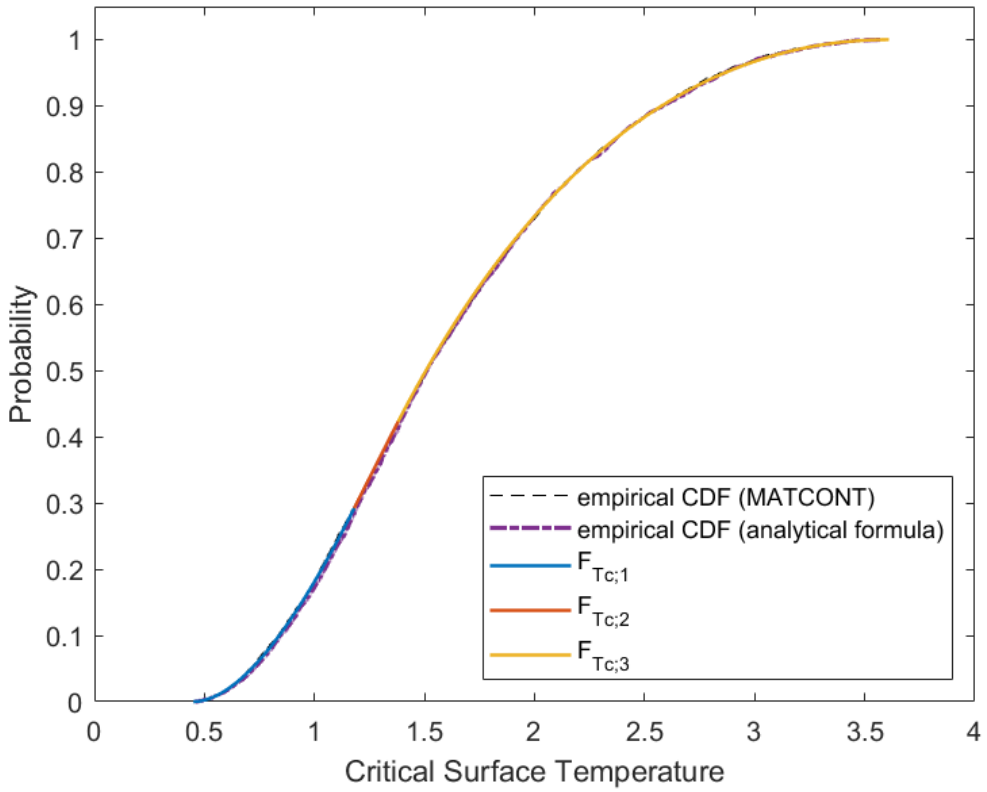


Figure 7.2: Cumulative distribution function of the critical surface temperature. The analytical result is depicted by the blue ($F_{T_c;1}$), orange ($F_{T_c;2}$) and yellow ($F_{T_c;3}$) lines. The dashed black and purple lines are obtained from empirical data, each shows the empirical cumulative distribution function.

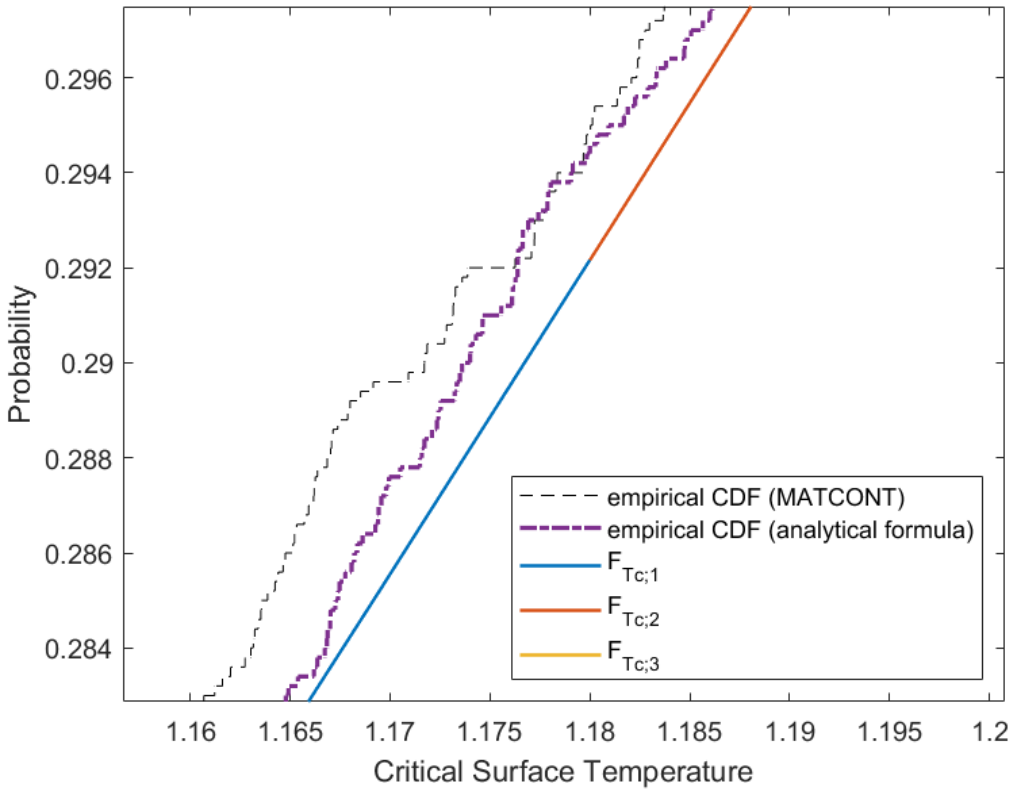


Figure 7.3: Zoom into the cumulative distribution function of the critical surface temperature.

In Fig. 7.2, the lines cannot completely be distinguished. Therefore, Fig. 7.3 shows a close up.

As we can see from Fig. 7.2, the analytical result follows both empirical CDFs pretty well. Furthermore, we can conclude that the piecewise functions are continuous.

7.3.3 Analysis of the Characteristics

Finally, both analytical and numerical data of the PDF of the critical surface temperature T_c from Section 7.2.2 and Section 7.3.1 are evaluated.

From Fig. 7.1, we see that the PDF is not symmetric and has a positive skew. The density function has a steep increase (see blue line; $f_{T_c;1}$), a slight constant part (see orange line; $f_{T_c;2}$), and on the right side a tail that decreases slowly (see yellow line; $f_{T_c;3}$).

These are only visual observations. In the course of this section, we determine further information from data and the analytical formula (see Equation (7.13)) mathematically.

As outlined in L. Fahrmeir et. al., PDFs can be described by certain characteristics. To gain insight into the PDF of the critical surface temperature, we have a look at the characteristics mean, variance, standard deviation, median and skewness. [14, Sec. 2.2]

The results are summarized in Table 7.1.

Characteristics	Empirical Data (`MATCONT`)	Empirical Data (analytical formula)	Analytical Solution
Mean E[T_c]	1.6155	1.6314	1.617053
Variance Var[T_c]	0.4244	0.4281	0.4272
Standard Deviation s	0.6514	0.6543	0.6536
Median z_m	1.5112	1.5245	1.50583
Skewness	0.5567	0.5564	-/-

Table 7.1: Characteristics of the probability density function of the critical surface temperature. The table compares the properties from the analytical and numerical results.

Numerical Results For each property there is a suitable `MATLAB` function. [44, 46,53,55,58] They can be used to determine the properties of the data obtained with `MATCONT` and the analytical function (see second and third column of Table 7.1).

Analytical Results The mean, variance, standard deviation, and median are calculated from the analytically calculated formula of the PDF (see Equation (7.13)) and the CDF (see Equation (7.14)). For this, `Mathematica` is used and the results are presented in the fourth column of Table 7.1.

The *mean*, or *expected value*, of the critical surface temperature is calculated using Definition 3.1 given in L. Wasserman. [76] The mean of piecewise functions is then the sum of the expected value of each section.

$$\begin{aligned}
\mathrm{E}[T_c] &= \int_{-\infty}^{\infty} z f_{T_c}(z)\,dz \\
&= \int_{7(\frac{0.72}{8})^{8/7}}^{7(\frac{1.68}{8})^{8/7}} z f_{T_c;1}(z)\,dz + \int_{7(\frac{1.68}{8})^{8/7}}^{7(\frac{1.92}{8})^{8/7}} z f_{T_c;2}(z)\,dz + \int_{7(\frac{1.92}{8})^{8/7}}^{7(\frac{4.48}{8})^{8/7}} z f_{T_c;3}(z)\,dz \\
&= 0.263147 + 0.161776 + 1.19213 \\
&= 1.617053
\end{aligned} \tag{7.16}$$

The *variance* for the random variable T_c is defined as [22, Def. 1.9.2]

$$\mathrm{Var}[T_c] = \mathrm{E}[T_c^2] - (\mathrm{E}[T_c])^2\,. \tag{7.17}$$

This intermediate calculation is necessary to receive the value of $\mathrm{E}[T_c^2]$.

$$\begin{aligned}
\mathrm{E}[T_c^2] &= \int_{-\infty}^{\infty} z^2 f_{T_c}(z)\,dz \\
&= \int_{7(\frac{0.72}{8})^{8/7}}^{7(\frac{1.68}{8})^{8/7}} z^2 f_{T_c;1}(z)\,dz + \int_{7(\frac{1.68}{8})^{8/7}}^{7(\frac{1.92}{8})^{8/7}} z^2 f_{T_c;2}(z)\,dz + \int_{7(\frac{1.92}{8})^{8/7}}^{7(\frac{4.48}{8})^{8/7}} z^2 f_{T_c;3}(z)\,dz \\
&= 0.248365 + 0.20632 + 2.58735 \\
&= 3.042035
\end{aligned} \tag{7.18}$$

Then, the individual results from Equation (7.16) and Equation (7.18) yield the variance of the critical surface temperature.

$$\mathrm{Var}[T_c] = \mathrm{E}[T_c^2] - (\mathrm{E}[T_c])^2 = 3.042035 - (1.617053)^2 = 0.4272 \tag{7.19}$$

The *standard deviation* is the square root of the variance. [76, Def. 3.14]

$$\mathrm{s} = \sqrt{\mathrm{Var}[T_c]} = \sqrt{0.4272} \approx 0.6536 \tag{7.20}$$

To determine the median of a function, we need its CDF. Then, the median x_m is given by solving $F_X(x_m) = \frac{1}{2}$. [22, Ex. 2.1.7]

As we can observe from Fig. 7.2, the probability $\frac{1}{2}$ is achieved for a certain critical surface temperature by the third piecewise function $F_{T_c;3}$. Thus, we use this information and compute the median $\mathrm{z_m}$ of the critical surface temperature with the help of `Mathematica`.

$$
\begin{aligned}
& F_{T_c;3}(\mathrm{z_m}) = \frac{1}{2} \\
& \Leftrightarrow {\mathrm{z_m}}^{7/8} \left(1.93388 - 0.797108 \ln(\mathrm{z_m})\right) - 1.8 = \frac{1}{2} \\
& \Rightarrow \mathrm{z_m} = 1.50583
\end{aligned}
\tag{7.21}
$$

Evaluation Comparing the individual columns of Table 7.1 with each other shows us that the results hardly differ from each other. From this we can conclude that our observation that the different representations of the PDFs in Fig. 7.1, namely piecewise functions and two histograms, coincide.

Another observation is that even though the input parameters γ and Γ are uniformly distributed, the PDF of the critical surface temperature is not uniform.

The expected value $\mathrm{E}[T_c]$ for all data obtained with the three methods lies at approximately 1.6. The median Z_m of all results deviates slightly from the mean, and is approximately 1.5.

The skewness is approximately 0.56 and the variance is approximately 0.42. These show that the form of the PDF is asymmetrical and the values are mostly distributed on the right tail. [14, Sec. 2.2]

7.3.4 Conclusion

According to the results from Table 7.1, there is a probability of around 60 % that the tipping point lies approximately at 1.6 °C. But there is also a possibility of 30 % that the melting of the Greenland ice sheet is triggered at 2.2 °C. More detailed interpretation and consequences will be left to climatologists.

Chapter 7 offers a framework for probabilistic analysis of the Greenland ice sheet model 4.1. With this, the PDF and CDF of the climate model were determined analytically and numerically. Thereby, our analysis was for the parameters of the model γ and Γ being uniformly distributed.

Hence, this framework serves as a basis for further probabilistic analyses. To get started, the PDF of the critical surface temperature T_c might be determined using different distributions for the parameters γ and Γ. Then, a detailed analysis between different PDFs could give more information to climatologists.

Furthermore, it is encouraged that the methods presented in this chapter be applied to other climate models as well.

8. Conclusion and Further Work

The motivation of this thesis was to analyze the impacts of parameter uncertainties of the Greenland ice sheet Model 4.1 through sensitivity and probabilistic analyses and to provide a `MATLAB` framework for each of these.

In Chapter 4, numerical continuation was applied to the Greenland ice sheet model (Model 4.1). In doing so, we compared the different methods `vpasolve`, `fsolve`, and `MATCONT`. The comparison was based on implementation results, obstacles, and performance benchmarking. We have seen that `MATCONT` is the most suitable package. This is not only because the bifurcation diagram was displayed without any problems, but also because the benchmark showed `MATCONT` to be significantly faster than the other methods.

For bifurcation analysis, the `MATLAB` function `bifurcation_points` was implemented. This was used throughout the thesis for further analyses of Model 4.1.

With local and global sensitivity analysis, one can among others determine the sensitivity of individual parameters and also the sensitivity of the interaction between parameters. [36, Sec. 6.5.1] [3, 79] The global sensitivity analysis, was performed with the Sobol method.

We then established a sensitivity analysis framework in Section 5.3 to analyze the sensitivity of the Greenland ice sheet model. The result of the local sensitivity analysis of Model 4.1 was that both parameters are sensitive and have an influence on the outcome (see Fig. 5.4). For the global sensitivity analysis, the application of the Sobol method to the critical temperature showed that both parameters have a comparable influence on the outcome of Model 4.1 (see Table 5.2 and Fig. 5.5). We also analyzed the global sensitivity of the set of the critical temperatures, which yielded results similar to the previous analysis (see Table 5.3 and Fig. 5.8).

Besides the sensitivity analysis framework in Section 5.3, a novel `MATLAB` function `sobol_method` was created as implementation of the Sobol method. This function is adaptable, such that the partial variances and Sobol indices of one-dimensional functions with any number of parameters can be numerically approximated.

Bifurcation and sensitivity analysis were then combined in Chapter 6, so that the global sensitivity of branches of equilibria could be analyzed. Therefore, we looked at two approaches: 1-Wasserstein-inspired distance and 1-Wasserstein distance. Both distances were then applied to the Sobol method and thus used to analyze the

parameter influence of simple one-dimensional ODEs.

Results from this analysis showed that the new approach to linking bifurcation and sensitivity analysis is viable and can be applied to climate models in the future.

The thesis was concluded with a probabilistic analysis of the critical surface temperature. Chapter 7 shows a structured way of analytically and numerically computing the PDF (as shown in Fig. 7.1) as well as the CDF (as shown in Fig. 7.2) of the critical surface temperature. The challenge was to determine the PDF of a random variable, that itself is a product of independent random variables. To solve this, the Mellin convolution (see Definition 7.2) and the transformation of continuous random variables (see Theorem 7.2) were applied.

The PDF of the critical surface temperature was then visually compared to the PDF obtained with empirical data from the numerical analysis. We could determine by visual observation that the analytical and numerical analyses agree. This was further reinforced by computations of the properties of the PDF, such as the mean and median.

Overall, different methods for analyzing parameter impacts on the Greenland ice sheet Model 4.1 were presented. These are sensitivity analysis, sensitivity analysis using Wasserstein distances and probabilistic analysis. Each can be utilized to enhance the quality of ice sheet models.

The thesis presents a foundational framework based on the Greenland ice sheet model (Model 4.1), for which the parameters of the critical surface temperature turned out to be sensitive. These analyses were done for an assumed uniform distribution of the parameters.

For continuing these analyses, observational data might be used to estimate the PDFs for the parameters and then to apply them on the presented frameworks. These frameworks also can be applied to other ice sheet models with only making minor adjustments due to the implementations for the bifurcation and sensitivity analysis being kept general.

Bibliography

[1] William A. Adkins and Mark G. Davidson. *Ordinary Differential Equations*. Springer, 2012.

[2] C. Donald Ahrens and Robert Henson. *Meteorology today: an introduction to weather, climate, and the environment*. Cengage learning, 2009.

[3] GEB Archer, Andrea Saltelli, and Ilya Meyerovich Sobol. Sensitivity Measures, ANOVA-like Techniques and the Use of Bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120, 1997.

[4] Earth Science Communications Team at NASA's Jet Propulsion Laboratory. Global Temperature. `https://climate.nasa.gov/vital-signs/global-temperature`, 2022. Retrieved: June 04, 2022.

[5] Niklas Boers and Martin Rypdal. Critical slowing down suggests that the western Greenland Ice Sheet is close to a tipping point. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

[6] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

[7] Flavio Cannavó. Sensitivity analysis for volcanic source modeling quality assessment and model selection. *Computers & Geosciences*, 44:52–59, 2012.

[8] Carmen Chicone. *Ordinary Differential Equations with Applications*, volume 34. Springer, Second edition, 2006.

[9] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. SIAM, 2000.

[10] Philip J. Davis and Philip Rabinowitz. *Methods of numerical integration*. Academic Press, 1984.

[11] Robert P. Dickinson and Robert J. Gelinas. Sensitivity Analysis of Ordinary Differential Equation Systems — A Direct Method. *Journal of Computational Physics*, 21(2):123–143, 1976.

[12] Henk A. Dijkstra. Numerical bifurcation methods applied to climate models: analysis beyond simulation. *Nonlinear Processes in Geophysics*, 26(4):359–369, 2019.

[13] Henk A. Dijkstra, Fred W. Wubs, Andrew K. Cliffe, Eusebius Doedel, Ioana F. Dragomirescu, Bruno Eckhardt, Alexander Yu Gelfgat, Andrew L. Hazel, Valerio Lucarini, Andy G. Salinger, Erik T. Phipps, Juan Sanchez-Umbria, Henk Schuttelaars, Laurette S. Tuckerman, and Uwe Thiele. Numerical Bifurcation Methods and their Application to Fluid Dynamics: Analysis beyond Simulation. *Communications in Computational Physics*, 15(1):1–45, 2014.

[14] Ludwig Fahrmeir, Christian Heumann, Rita Künstler, Iris Pigeot, and Gerhard Tutz. *Statistik: Der Weg zur Datenanalyse. (German) [Statistics: The Path to Data Analysis]*. Springer, Eighth edition, 2016.

[15] Valerio Faraoni. Lagrangian formulation, a general relativity analogue, and a symmetry of the Vialov equation of glaciology. *The European Physical Journal Plus*, 135(11):1–12, 2020.

[16] Alessio Figalli and Federico Glaudo. *An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows*. European Mathematical Society, 2021.

[17] Andrew G. Glen, Lawrence M. Leemis, and John H. Drew. Computing the distribution of the product of two continuous random variables. *Computational Statistics & Data Analysis*, 44(3):451–464, 2004.

[18] W. Govaerts. Numerical bifurcation analysis for ODEs. *Journal of Computational and Applied Mathematics*, 125(1-2):57–68, 2000.

[19] W. Govaerts, Yu. A. Kuznetsov, H.G.E. Meijer, B. Al-Hdaibat, V. De Witte, A. Dhooge, W. Mestrom, N. Neirynck, A.M. Riet, and B. Sautois. MATCONT: Continuation toolbox for ODEs in Matlab. *U. Gent*, 2018.

[20] Willy J. F. Govaerts. *Numerical Methods for Bifurcations of Dynamical Equilibria*. SIAM, 2000.

[21] John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42. Springer, 1983.

[22] Robert V. Hogg, Joseph W. McKean, and Allen T. Craig. *Introduction to Mathematical Statistics*. Pearson, Eighth edition, 2019.

[23] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17, 1996.

[24] J. T. Houghton, Y. Ding, D. J. Griggs, M. Noguer, P. J. van der Linden, X. Dai, K. Maskell, and C. A. Johnson. *Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2001. Intergovernmental Panel on Climate Change IPCC 2001.

[25] IPCC. IPCC - Intergovernmental Panel on Climate Change. https://www.ipcc.ch/, 2022. Retrieved: September 21, 2022.

[26] Yoshihiko Iseri, Sayaka Yoshikawa, Masashi Kiguchi, Ryunosuke Tawatari, Shinjiro Kanae, and Taikan Oki. Towards the incorporation of tipping elements in global climate risk management: probability and potential impacts of passing a threshold. *Sustainability science*, 13(2):315–328, 2018.

[27] T. Ishigami and T. Homma. An Importance Quantification Technique in Uncertainty Analysis for Computer Models. In *[1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, pages 398–403. IEEE, 1990.

[28] Achim Klenke. *Wahrscheinlichkeitstheorie. (German) [Probability Theory]*, volume 3. Springer, 2013.

[29] Niklas Kolbe. Wasserstein distance. https://github.com/nklb/wasserstein-distance, 2022. Retrieved: July 04, 2022.

[30] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K. Rohde. Optimal Mass Transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

[31] Konrad Königsberger. *Analysis 1. (German) [Analysis 1]*. Springer, 2004.

[32] Yuri A. Kuznetsov. *Elements of Applied Bifurcation Theory*, volume 112. Springer, Third edition, 2004.

[33] Timothy M. Lenton, Hermann Held, Elmar Kriegler, Jim W. Hall, Wolfgang Lucht, Stefan Rahmstorf, and Hans Joachim Schellnhuber. Tipping elements in the Earth's climate system. *Proceedings of the national Academy of Sciences*, 105(6):1786–1793, 2008.

[34] Anders Levermann and Ricarda Winkelmann. A simple equation for the melt elevation feedback of ice sheets. *The Cryosphere*, 10(4):1799–1807, 2016.

[35] Kerstin Lux, Peter Ashwin, Richard Wood, and Christian Kuehn. Uniting Parametric Uncertainty and Tipping Diagrams. *arXiv preprint arXiv:2110.15859*, 2021.

[36] Maia Martcheva. *An Introduction to Mathematical Epidemiology*, volume 61. Springer, 2015.

[37] MathWorks. cdfplot. https://de.mathworks.com/help/stats/cdfplot.html, 2022. Retrieved: September 13, 2022.

[38] MathWorks. Equation Solving Algorithms. https://de.mathworks.com/help/optim/ug/equation-solving-algorithms.html, 2022. Retrieved: May 04, 2022.

[39] MathWorks. format. https://de.mathworks.com/help/matlab/ref/format.html, 2022. Retrieved: June 14, 2022.

[40] MathWorks. fsolve. https://de.mathworks.com/help/optim/ug/fsolve.html, 2022. Retrieved: May 01, 2022.

[41] MathWorks. histogram. https://de.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram.html, 2022. Retrieved: July 28, 2022.

[42] MathWorks. load. https://de.mathworks.com/help/matlab/ref/load.html, 2022. Retrieved: June 08, 2022.

[43] MathWorks. makedist. https://de.mathworks.com/help/stats/makedist.html, 2022. Retrieved: June 08, 2022.

[44] MathWorks. mean. https://de.mathworks.com/help/matlab/ref/mean.html, 2022. Retrieved: September 13, 2022.

[45] MathWorks. Measure the Performance of Your Code. https://de.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html, 2022. Retrieved: June 24, 2022.

[46] MathWorks. median. https://de.mathworks.com/help/matlab/ref/median.html, 2022. Retrieved: September 13, 2022.

[47] MathWorks. nchoosek. https://de.mathworks.com/help/symbolic/nchoosek.html, 2022. Retrieved: May 28, 2022.

[48] MathWorks. net. https://de.mathworks.com/help/stats/haltonset.net.html?searchHighlight=net&s_tid=srchtitle_net_1, 2022. Retrieved: May 24, 2022.

[49] MathWorks. polyfit. https://de.mathworks.com/help/matlab/ref/polyfit.html, 2022. Retrieved: June 14, 2022.

[50] MathWorks. random. https://de.mathworks.com/help/stats/prob.normaldistribution.random.html, 2022. Retrieved: June 08, 2022.

[51] MathWorks. save. https://de.mathworks.com/help/matlab/ref/save.html, 2022. Retrieved: June 08, 2022.

[52] MathWorks. Save and Load Parts of Variables in MAT-Files. https://de.mathworks.com/help/matlab/import_export/load-parts-of-variables-from-mat-files.html, 2022. Retrieved: June 08, 2022.

[53] MathWorks. skewness. https://de.mathworks.com/help/stats/skewness.html, 2022. Retrieved: September 17, 2022.

[54] MathWorks. sobolset. https://de.mathworks.com/help/stats/sobolset.html, 2022. Retrieved: May 24, 2022.

[55] MathWorks. std. https://de.mathworks.com/help/matlab/ref/std.html, 2022. Retrieved: September 17, 2022.

[56] MathWorks. struct. https://de.mathworks.com/help/matlab/ref/struct.html, 2022. Retrieved: May 10, 2022.

[57] MathWorks. timeit. https://de.mathworks.com/help/matlab/ref/timeit.html, 2022. Retrieved: June 24, 2022.

[58] MathWorks. var. https://de.mathworks.com/help/matlab/ref/var.html, 2022. Retrieved: September 17, 2022.

[59] MathWorks. vpasolve. https://de.mathworks.com/help/symbolic/sym.vpasolve.html, 2022. Retrieved: May 01, 2022.

[60] Slashdot Media. MatCont. Numerical Bifurcation Analysis Toolbox in Matlab. https://sourceforge.net/projects/matcont/, 2022. Retrieved: May 05, 2022.

[61] Mark F. Meier. Flow of the ice sheets. https://www.britannica.com/science/glacier/Net-mass-balance#ref65687, 2022. Retrieved: July 04, 2022.

[62] H.-O. Pörtner, D.C. Roberts, V. Masson-Delmotte, P. Zhai, M. Tignor, E. Poloczanska, K. Mintenbeck, A. Alegría, M. Nicolai, A. Okem, J. Petzold, B. Rama, and N.M. Weyer (eds.). IPCC Special Report on the Ocean and Cryosphere in a Changing Climate. *IPCC 2019*, 2019.

[63] MathWorld-A Wolfram Web Resource. Discrepancy. https://mathworld.wolfram.com/Discrepancy.html, 2022. Retrieved: June 16, 2022.

[64] Romain Richard, Jérôme Casas, and Edward McCauley. Sensitivity analysis of continuous-time models for ecological and evolutionary theories. *Theoretical Ecology*, 8(4):481–490, 2015.

[65] I. M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[66] Alastair Spence and Ivan G. Graham. Numerical Methods for Bifurcation Problems. In *The Graduate Student's Guide to Numerical Analysis' 98*, pages 177–216. Springer, 1999.

[67] Melvin Dale Springer. *The Algebra of Random Variables*. John Wiley & Sons, 1979.

[68] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC press, 2015.

[69] Timothy John Sullivan. *Introduction to Uncertainty Quantification*, volume 63. Springer, 2015.

[70] TiPES. About. `https://www.tipes.dk/about/`, 2022. Retrieved: September 21, 2022.

[71] TiPES. TiPES - Tipping Points in the Earth System. `https://www.tipes.dk/`, 2022. Retrieved: September 21, 2022.

[72] TiPES. Tipping Points. `https://www.tipes.dk/tipping-points/`, 2022. Retrieved: September 21, 2022.

[73] George A.K. Van Voorn and Bob W. Kooi. Combining bifurcation and sensitivity analysis for ecological models. *The European Physical Journal Special Topics*, 226(9):2101–2118, 2017.

[74] Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2009.

[75] Wolfgang Walter. *Gewöhnliche Differentialgleichungen: Eine Einführung. (German) [Ordinary Differential Equations: An Introduction]*. Springer, 2000.

[76] Larry Wasserman. *All of Statistics. A Concise Course in Statistical Inference*. Springer, 2004.

[77] Wu Hsiung Wu, Feng Sheng Wang, and Maw Shang Chang. Dynamic sensitivity analysis of biological systems. *BMC Bioinformatics*, 9(12):1–17, 2008.

[78] Jianzhong Zhang and Chengxian Xu. Trust region dogleg path algorithms for unconstrained minimization. *Annals of Operations Research*, 87:407–418, 1999.

[79] X.-Y. Zhang, M. N. Trame, L. J. Lesko, and S. Schmidt. Sobol Sensitivity Analysis: A Tool to Guide the Development and Evaluation of Systems Pharmacology Models. *CPT: Pharmacometrics & Systems Pharmacology*, 4(2):69–79, 2015.

A. Appendix

A.1 Numerical Continuation

```
n = 250;
t = linspace(0,1,n);
y = zeros(n,2);

for i = 1:250
  L = t(i);
  syms x;
  S = vpasolve(-x.^2 + L == 0, x);
  y(i,:) = S;
end

plot(t,y)
ylabel('x')
xlabel('\lambda')
```

Listing A.1: Numerical continuation of Example 3.1.

```
function [y, greatest_temp] = vpasolve_bifurcation(
    gamma, Gamma)
  n = 500;
  T = linspace(0, 5, n);

  y = zeros(n,2);
  greatest_temp = 0;

  for i = 1:n
      syms h;
      S = vpasolve(-h.^8 + gamma*Gamma*h - T(i) == 0, h
          , [0 Inf]);

      if size(S,1) == 2
```

```
13              y(i,:) = S;
14              greatest_temp = T(i);
15          else
16              y(i,:) = [NaN NaN];
17          end
18      end
19  end
```

Listing A.2: Numerical continuation of the Greenland ice sheet model (Model 4.1) using `vpasolve`.

```
1  c = colors();
2  n = 500;
3  gamma = 4.4*100;
4  Gamma = 5/1000;
5  T = linspace(0, 5, n);
6
7  [y, greatest_temp] = vpasolve_bifurcation(gamma,Gamma);
8
9  plot(T, y, 'Color', c('blue'))
10 hold on;
11 plot(greatest_temp, ((gamma*Gamma/8).^(1/7)), '*', '
       Color', c('orange'))
12 text(greatest_temp + 0.02, ((gamma*Gamma/8).^(1/7)), '
       LP');
13 xlabel('Surface Temperature T');
14 ylabel('Ice Thickness h');
15 ylim([0 1.2]);
16 xlim([0 2]);
17
18 figure(2); % Zoom-In
19 plot(T, y, 'Color', c('blue'))
20 hold on;
21 plot(greatest_temp, ((gamma*Gamma/8).^(1/7)), '*', '
       Color', c('orange'))
22 text(greatest_temp + 0.0002, ((gamma*Gamma/8).^(1/7)),
       'LP');
23 xlabel('Surface Temperature T');
24 ylabel('Ice Thickness h');
25 ylim([0.75 0.9]);
26 xlim([1.59 1.6]);
```

Listing A.3: Bifurcation diagram of the Greenland ice sheet model (Model 4.1) using `vpasolve`.

```
function [y, greatest_temp] = fsolve_bifurcation(gamma,
    Gamma, initial_point)
  n = 500;
  T = linspace(0, 5, n);

  y = zeros(n,2);
  greatest_temp = 0;

  options = optimoptions(@fsolve,'StepTolerance',1e-20,
      ...
      'FunctionTolerance',1e-20,'MaxFunctionEvaluations
        ',1e5, ...
      'MaxIterations',1e5,'Display','iter');

  for i = 1:n
      t = T(i);
      greatest_temp = t;

      gis_model = @(h)[real(-h.^8 + Gamma.*gamma.*h - t
        )];

      syms h;
      y(i,:) = fsolve(gis_model, initial_point, options
        );
  end
end
```

Listing A.4: Numerical continuation of the Greenland ice sheet model (Model 4.1) using `fsolve`.

```
c = colors();
n = 500;
gamma = 4.4*100;
Gamma = 5/1000;
T = linspace(0, 5, n);

[y, greatest_temp] = fsolve_bifurcation(gamma, Gamma,
   [0.1 1]);

figure(1)
plot(T, y, 'Color', c('blue'))
hold on;
plot(greatest_temp, ((gamma*Gamma/8).^(1/7)), '*', '
```

```
    Color', c('orange'))
13 text(greatest_temp + 0.02, ((gamma*Gamma/8).^(1/7)), '
    LP');
14 xlabel('Surface Temperature T');
15 ylabel('Ice Thickness h');
16 xlim([0 2]); ylim([0 1.2]);
17
18 [y, greatest_temp] = fsolve_bifurcation(gamma, Gamma,
    [0 0]);
19
20 figure(2);
21 plot(T,y, 'Color', c('blue'))
22 hold on;
23 plot(greatest_temp, ((gamma*Gamma/8).^(1/7)), '*', '
    Color', c('orange'))
24 text(greatest_temp + 0.02, ((gamma*Gamma/8).^(1/7)), '
    LP');
25 xlabel('Surface Temperature T');
26 ylabel('Ice Thickness h');
27 xlim([0 2]); ylim([0 1.2]);
```

Listing A.5: Bifurcation diagram of the Greenland ice sheet model (Model 4.1) using `fsolve`.

```
1 bf = bifurcation_points(4.4*100, 5/1000);
2 cpl(bf.x, bf.v, bf.s, [2 1])
3 xlim([0 2]);
4 xlabel('Surface Temperature T');
5 ylabel('Ice Thickness h');
```

Listing A.6: Bifurcation diagram of the Greenland ice sheet model (Model 4.1) using `MATCONT`.

```
1 function bfpts = bifurcation_points_for_timeit(gamma,
    Gamma)
2     p = [gamma; Gamma; 0];
3     ap = 3;
4     [x0,v0] = init_EP_EP(@IceSheet_LW,0,p,ap);
5     opt = contset;
6     opt = contset(opt,'MaxNumPoints',500);
7     opt = contset(opt,'Singularities',1);
8     [x,v,s,h,f] = cont(@equilibrium,x0,[],opt);
```

```
9
10      bfpts = struct('gamma',gamma,'Gamma',Gamma,'x',x,'v
            ',v,'s',s);
11  end
```

Listing A.7: Numerical continuation of Model 4.1, adjusted for performance benchmarking.

```
1  gamma = 440;
2  Gamma = 0.005;
3
4  vpasolve_bifurcation_timeit = @() vpasolve_bifurcation(
       gamma,Gamma);
5  fsolve_bifurcation_timeit = @() fsolve_bifurcation(
       gamma,Gamma);
6  bifurcation_points_timeit = @()
       bifurcation_points_for_timeit(gamma,Gamma);
7
8  timeit(vpasolve_bifurcation_timeit)
9  timeit(fsolve_bifurcation_timeit)
10 timeit(bifurcation_points_timeit)
```

Listing A.8: Benchmark analysis of the numerical continuation functions `vpasolve_bifurcation`, `fsolve_bifurcation` and `bifurcation_points`.

A.2 Sensitivity Analysis

```
1  c = colors();
2  n = 150;
3  t = linspace(0,4,n);
4  ode_ex = @(t,p) 2.*t.*exp(p*t);
5
6  plot(t, ode_ex(t,-2.1), 'LineStyle', '--', 'Color', c('
       orange'), 'LineWidth',0.8)
7  hold on;
8  plot(t, ode_ex(t, -2), 'LineStyle', '-', 'Color', c('
       blue'), 'LineWidth',0.8)
9  hold on;
10 plot(t, ode_ex(t, -1.9), 'LineStyle', '-.', 'Color', c(
       'yellow'), 'LineWidth',0.8);
11 hold on;
12 xlabel('t')
13 ylabel('S_p(t)')
```

```
14 legend('p = -2.1', 'p = -2.0', 'p = -1.9')
```

Listing A.9: Local sensitivity analysis of Example 5.1.

```
1 function y = sensitivity_lapserate(t, gamma, Gamma,
     Temp)
2       y = exp(gamma*Gamma*t)/gamma*t - Temp/(gamma*Gamma
          ^2);
3 end
```

Listing A.10: Implementation of Equation (5.30).

```
1 function y = sensitivity_melting(t, gamma, Gamma, Temp)
2       y = exp(gamma*Gamma*t)/Gamma*t - Temp/(Gamma*gamma
          ^2);
3 end
```

Listing A.11: Implementation of Equation (5.31).

```
1  n = 500;
2  m = 5;
3  t = linspace(0.5,4,n);
4
5  gamma = [4.2; 4.3; 4.4; 4.5; 4.6]*100;
6  Gamma = [4.8; 4.9; 5; 5.1; 5.2]/1000;
7  Temp = 1.6;
8
9  M = ones(n,m);
10 L = ones(n,m);
11
12 for i = 1:n
13     for j = 1:m
14         M(i,j) = sensitivity_melting(t(i), gamma(j),
               5/1000, Temp);
15         L(i,j) = sensitivity_lapserate(t(i), 4.4*100,
               Gamma(j), Temp);
16     end
17 end
18
19 linS = {'-','--','-.','--', '-'};
20
```

```
21 subplot(1,2,1);
22 for i = 1:m
23     plot(t, M(:,i), 'linestyle', linS{i})
24     hold on;
25 end
26 legend('\gamma = 4.2\cdot10^2', '\gamma = 4.3\cdot10^2'
      , '\gamma = 4.4\cdot10^2', '\gamma = 4.5\cdot10^2',
      '\gamma = 4.6\cdot10^2');
27 title('Sensitivity of \gamma');
28 xlabel('t');
29 ylabel('S_{\gamma}(t)');
30
31 subplot(1,2,2);
32 for i = 1:m
33     plot(t, L(:,i), 'linestyle', linS{i})
34     hold on;
35 end
36 legend('\Gamma = 4.8\cdot10^-3', '\Gamma = 4.9\cdot10
      ^-3', '\Gamma = 5.0\cdot10^-3', '\Gamma = 5.1\cdot10
      ^-3', '\Gamma = 5.2\cdot10^-3');
37 title('Sensitivity of \Gamma');
38 xlabel('t');
39 ylabel('S_{\Gamma}(t)');
```

Listing A.12: Local sensitivity analysis of the Greenland ice sheet model (Model 4.1).

```
1  n = 2;
2  N = 2^11;
3
4  p = sobolset(3*n, 'skip', 1e3);
5  x = net(p,N);
6
7  lower_bounds = [240 0.003];
8  upper_bounds = [640 0.007];
9
10 matrix_current = x(:,1:n).*(upper_bounds - lower_bounds
      )+lower_bounds;
11 matrix_sample = x(:,(n+1):2*n).*(upper_bounds -
      lower_bounds)+lower_bounds;
12 matrix_resample = x(:,(2*n+1):(3*n)).*(upper_bounds -
      lower_bounds)+lower_bounds;
13
```

```
[D, S] = sobol_method(n, N, @critical_temperature,
    matrix_current, matrix_sample, matrix_resample)

X = categorical({'\gamma','\Gamma', '\gamma \Gamma'});
X = reordercats(X,{'\gamma','\Gamma', '\gamma \Gamma'})
    ;
bar(X, [S(1,1), S(1,1)+S(1,2); S(2,1), S(2,1)+S(1,2); S
    (1,2) 0]);
xlabel('Parameters');
ylabel('(Total) Sobol Sensitivity Indices');
legend('$$\hat{S}$$','$$\hat{S}_T$$','Interpreter','
    Latex')
ylim([0 1]);
```

Listing A.13: Preparation, execution and visualization of the Sobol method for the critical temperature of the Greenland ice sheet model (Model 4.1).

```
rng('default');
n = 5000;

pd_g = makedist('Uniform',240,640);
pd_G = makedist('Uniform',0.003,0.007);

B = [];

for i = 1:n
    g = random(pd_g);
    G = random(pd_G);
    bp = bifurcation_points(g,G);
    B = [B; bp];
end

file_name = 'bifurcation_uniform_uniform.mat';
save(file_name, 'B');
```

Listing A.14: Creation of a `.mat` file with the points for branches of equilibria for the Greenland ice sheet model (Model 4.1). The parameters are drawn from a uniform distribution.

```
c = colors();
% for B, see Lst. A.14
n = size(B,1);

for i = 1:n
    plot(B(i).x(2,:),B(i).x(1,:),'Color',c('blue'))
    hold on;
    LP = B(i).s(2).index;
    plot(B(i).x(2,LP), B(i).x(1,LP), 'Color', c('orange
        '), 'Marker', '.')
end
xlim([0,4.5]);
xlabel('Surface Temperature');
ylabel('Ice Thickness');
```

Listing A.15: Visualization of all bifurcation diagrams with their respective bifurcation points.

```
rng('default')
s = 500;

pd_g = makedist('Uniform', 2.4*100, 6.4*100);
pd_G = makedist('Uniform', 0.003, 0.007);
g = sort(random(pd_g,s,1));
G = sort(random(pd_G,s,1));

Tc = [];
hc = [];

for i = 1:s
    T = 7*((g(i)*G(i))/8)^(8/7);
    h = ((g(i)*G(i))/8)^(1/7);
    hc = [hc; h];
    Tc = [Tc; T];
end

p = polyfit(Tc,hc,5)
```

Listing A.16: Polynomial regression for the set of critical points.

```
c = colors();
format('default')
rng('default')

% for B, see Lst. A.14
n = size(B,1);

s = 100;
pd_g = makedist('Uniform', 2.4*100, 6.4*100);
pd_G = makedist('Uniform', 0.003, 0.007);
g = sort(random(pd_g,s,1));
G = sort(random(pd_G,s,1));

fun = @(x)  0.0013*x.^5 -0.0156*x.^4 +0.0753*x.^3
   -0.1920*x.^2 +0.3127*x +0.6025;

y = [];
Tc = [];
hc = [];
for i = 1:s
    T = 7*((g(i)*G(i))/8)^(8/7);
    h = ((g(i)*G(i))/8)^(1/7);
    Tc = [Tc; T];
    hc = [hc; h];
    y = [y; fun(T)];
end

subplot(1,2,1)
plot(Tc, hc, 'o')
hold on;
plot(Tc, fun(Tc), '-')
xlabel('Critical Surface Temperature');
ylabel('Critical Ice Thickness');
legend('Data', 'Approximated Function');

subplot(1,2,2);
times = linspace(Tc(1),Tc(end));
plot(times,zeros(s,1),'k-')
hold on
plot(Tc, hc-fun(Tc), '.', 'Color', c('blue'))
ylim([-0.1 0.1])
xlabel('Fitted Values');
ylabel('Residuals');
```

Listing A.17: Comparison of the polynomial regression to the critical points.

```
n = 2;
N = 2^11;

p = sobolset(3*n, 'skip', 1e3);
x = net(p,N);

lower_bounds = [240 0.003];
upper_bounds = [640 0.007];

matrix_current = x(:,1:n).*(upper_bounds - lower_bounds)+lower_bounds;
matrix_sample = x(:,(n+1):2*n).*(upper_bounds - lower_bounds)+lower_bounds;
matrix_resample = x(:,(2*n+1):(3*n)).*(upper_bounds - lower_bounds)+lower_bounds;

[D, S] = sobol_method(n, N, @critical_points, matrix_current, matrix_sample, matrix_resample)

X = categorical({'\gamma','\Gamma', '\gamma \Gamma'});
X = reordercats(X,{'\gamma','\Gamma', '\gamma \Gamma'});
bar(X, [S(1,1), S(1,1)+S(1,2); S(2,1), S(2,1)+S(1,2); S(1,2) 0]);
xlabel('Parameters');
ylabel('(Total) Sobol Sensitivity Indices');
legend('$$\hat{S}$$','$$\hat{S}_T$$','Interpreter','Latex')
ylim([0 1]);
```

Listing A.18: Preparation, execution and visualization of the Sobol method for thes set of critical points of the Greenland ice sheet model (Model 4.1).

A.3 Wasserstein Distance and Sensitivity Analysis

```
f_hat = @(x) 0.0013*x.^5 -0.0156*x.^4 +0.0753*x.^3 -0.1920*x.^2 +0.3127*x +0.6025;
% for B, see Lst. A.14
n = size(B,1);
LP = zeros(n,4);
x = linspace(0,4,100);

```

```
for i = 1:n
    LP_i = B(i).s(2).index;

    LP(i,1) = i;
    LP(i,2) = LP_i; % index of LP
    LP(i,3) = B(i).x(2,LP_i); % critical surface
       temperature
    LP(i,4) = B(i).x(1,LP_i); % ice thickness
end

LP_sorted = sortrows(LP,3);

a = LP_sorted(1,1);
b = LP_sorted(end,1);
plot(B(a).x(2,:), B(a).x(1,:), 'Color', c('blue'))
hold on;
plot(B(b).x(2,:), B(b).x(1,:), 'Color', c('blue'))
hold on;
plot(x, f_hat(x), 'Color', c('orange'))

xlim([0,4])
xlabel('Surface Temperature');
ylabel('Ice Thickness');

str1 = [strcat('\gamma =' , num2str(B(a).gamma)),
   strcat(', \Gamma =' , num2str(B(a).Gamma))];
str2 = [strcat('\gamma =' , num2str(B(b).gamma)),
   strcat(', \Gamma =' , num2str(B(b).Gamma))];
legend(str1, str2, 'Approximated Function f')
```

Listing A.19: Bifurcation diagrams for lowest and highest critical temperature with the interpolated function of all critical points.

```
function bfpts = bifurcation_quadratic(a)
    p = [a; 0];
    ap = 2;
    [x0,v0] = init_EP_EP(@toymodel_quadratic,0,p,ap);
    opt = contset;
    opt = contset(opt,'MaxNumPoints',100);
    opt = contset(opt,'Singularities',1);
    opt = contset(opt,'Backward',1);
    [x1,v1,s1,h1,f1] = cont(@equilibrium,x0,[],opt);
    opt = contset;
    opt = contset(opt,'MaxNumPoints',100);
```

```
12      opt = contset(opt,'Singularities',1);
13      [x2,v2,s2,h2,f2] = cont(@equilibrium,x0,[],opt);
14      bfpts = struct('a',a,'x1',x1,'x2',x2,'v1',v1,'v2',
            v2,'s1',s1,'s2',s2);
15  end
```

Listing A.20: Numerical continuation of Model 6.1.

```
1  function y = toy_model_quadratic(x)
2      a = x(:,1);
3      y = (1/3)*(0.5-a)-1;
4  end
```

Listing A.21: Implementation of Model 6.1.

```
1  c = colors();
2  C = {c('blue'),c('orange'),c('yellow'),c('purple'),c('
       green')};
3  parameter = linspace(1,5,5);
4
5  for i = 1:5
6      a = parameter(i);
7      bfpts = bifurcation_quadratic(a);
8      plot(bfpts.x1(2,:), bfpts.x1(1,:), 'color', C{i});
9      hold on;
10     plot(bfpts.x2(2,:), bfpts.x2(1,:), 'color', C{i});
11     hold on;
12 end
13
14 a = 0.5;
15 bfpts = bifurcation_quadratic(a);
16 plot(bfpts.x1(2,:), bfpts.x1(1,:), 'color', c('
       lightblue'), 'LineStyle', '--');
17 hold on;
18 plot(bfpts.x2(2,:), bfpts.x2(1,:), 'color', c('
       lightblue'), 'LineStyle', '--');
19
20 legend('a=1','','a=2','','a=3','','a=4','','a=5','','
       reference')
21 xlabel('\lambda')
22 ylabel('x')
```

Listing A.22: Bifurcation diagrams of Model 6.1.

```
function bfpts = bifurcation_linear(b)
    p = [b; -0.1];
    ap = 2;
    [x0,v0] = init_EP_EP(@toymodel_linear,0,p,ap);
    opt = contset;
    opt = contset(opt,'MaxNumPoints',100);
    opt = contset(opt,'Singularities',1);
    opt = contset(opt,'Backward',1);
    [x1,v1,s1,h1,f1] = cont(@equilibrium,x0,[],opt);
    opt = contset;
    opt = contset(opt,'MaxNumPoints',100);
    opt = contset(opt,'Singularities',1);
    [x2,v2,s2,h2,f2] = cont(@equilibrium,x0,[],opt);
    bfpts = struct('b',b,'x1',x1,'x2',x2,'v1',v1,'v2',
       v2,'s1',s1,'s2',s2);
end
```

Listing A.23: Numerical continuation of Model 6.2.

```
function y = toy_model_linear(x)
    b = x(:,1);
    y = 0.5*b-0.25;
end
```

Listing A.24: Implementation of Model 6.2.

```
c = colors();
C = {c('blue'),c('orange'),c('yellow'),c('purple'),c('
   green')};
parameter = linspace(1,5,5);

for i = 1:5
    b = parameter(i);
    bfpts = bifurcation_linear(b);
    plot(bfpts.x1(2,:), bfpts.x1(1,:), 'color', C{i});
    hold on;
    plot(bfpts.x2(2,:), bfpts.x2(1,:), 'color', C{i});
    hold on;
end

b = 0.5;
```

```
15 bfpts = bifurcation_linear(b);
16 plot(bfpts.x1(2,:), bfpts.x1(1,:), 'color', c('
      lightblue') ,'LineStyle', '--');
17 hold on;
18 plot(bfpts.x2(2,:), bfpts.x2(1,:), 'color', c('
      lightblue'), 'LineStyle', '--');
19
20 legend('b=1','','b=2','','b=3','','b=4','','b=5','','
      reference')
21 xlabel('\lambda')
22 ylabel('x')
```

Listing A.25: Bifurcation diagrams of Model 6.2.

```
1  rng('default');
2  n = 1;
3  N = 2^11;
4
5  p = sobolset(3*n, 'skip', 1e3);
6  x = net(p,N);
7
8  lower_bounds = 1;
9  upper_bounds = 5;
10
11 matrix_current = x(:,1:n).*(upper_bounds - lower_bounds
      )+lower_bounds;
12 matrix_sample = x(:,(n+1):2*n).*(upper_bounds -
      lower_bounds)+lower_bounds;
13 matrix_resample = x(:,(2*n+1):(3*n)).*(upper_bounds -
      lower_bounds)+lower_bounds;
14
15 [Dq_insp, Sq_insp] = sobol_method(n, N, @
      toy_model_quadratic, matrix_current, matrix_sample,
      matrix_resample)
16 [Dl_insp, Sl_insp] = sobol_method(n, N, @
      toy_model_linear, matrix_current, matrix_sample,
      matrix_resample)
17
18 save('toymodel_wasserstein_inspired.mat','Dq_insp','
      Dl_insp')
```

Listing A.26: Preparation and execution of the Sobol method for Model 6.1 and Model 6.2 using the 1-Wasserstein-inspired distance derived in Equation (6.9) and Equation (6.15).

```
rng('default');
format longE;
n = 1;
N = 2^11;

p = sobolset(3*n, 'skip', N);
x = net(p,N);

lower_bounds = 1;
upper_bounds = 5;

matrix_current = x(:,1:n).*[upper_bounds - lower_bounds]+lower_bounds;

a = 0.5;
bf_reference_q = bifurcation_quadratic(a).x2(1,:);

distance_vector_q = zeros(N,1);
for i = 1:N
    distance_vector_q(i) = ws_distance(bf_reference_q, bifurcation_quadratic(matrix_current(i,1)).x2(1,:),1);
end

f0_q = sum(distance_vector_q)./N
Dq = sum(distance_vector_q.^2)./N - f0_q.^2

b = 0.5;
bf_reference_l = bifurcation_linear(b).x2(1,:);

distance_vector_l = zeros(N,1);
for i = 1:N
    distance_vector_l(i) = ws_distance(bf_reference_l, bifurcation_linear(matrix_current(i,1)).x2(1,:),1);
end

f0_l = sum(distance_vector_l)./N
Dl = sum(distance_vector_l.^2)./N - f0_l.^2

save('toymodel_wasserstein.mat','Dq','Dl')
```

Listing A.27: Preparation and execution of the Sobol method for Model 6.1 and Model 6.2 using the 1-Wasserstein distance.

```
load('toymodel_wasserstein.mat')
load('toymodel_wasserstein_inspired.mat')

subplot(1,2,1)
X = categorical({'Quadratic Influence', 'Linear
    Influence'});
X = reordercats(X,{'Quadratic Influence', 'Linear
    Influence'});
bar(X, [Dq_insp; Dl_insp]);
ylabel('Variance obtained with Wasserstein-Inspired
    Distance')
ylim([0 0.4])

subplot(1,2,2)
X = categorical({'Quadratic Influence', 'Linear
    Influence'});
X = reordercats(X,{'Quadratic Influence', 'Linear
    Influence'});
bar(X, [Dq; Dl]);
ylabel('Variance obtained with Wasserstein Distance')
ylim([0 0.05])

figure(2)
X = categorical({'Quadratic Influence', 'Linear
    Influence'});
X = reordercats(X,{'Quadratic Influence', 'Linear
    Influence'});
bar(X, [Dq_insp; Dl_insp]);
ylabel('Variance obtained with Wasserstein-Inspired
    Distance')
ylim([0 0.4])

figure(3)
X = categorical({'Quadratic Influence', 'Linear
    Influence'});
X = reordercats(X,{'Quadratic Influence', 'Linear
    Influence'});
bar(X, [Dq; Dl]);
ylabel('Variance obtained with Wasserstein Distance')
ylim([0 0.05])
```

Listing A.28: Visualization of the variance obtained with 1-Wasserstein-inspired distance and 1-Wasserstein distance.

A.4 Probabilistic Analysis

```
c = colors();

z1 = linspace(0.45,1.18,70);
z2 = linspace(1.18,1.38,70);
z3 = linspace(1.38,3.61,70);

gg = random(makedist('Uniform',240,640),5000,1);
GG = random(makedist('Uniform',0.003,0.007),5000,1);
TT = 7*((gg.*GG)/8).^(8/7);

n = 70;
% for B, see Lst. A.14
m = size(B,1);

Tc = [];
for i = 1:m
    LP = B(i).s(2).index;
    Tc = [Tc; B(i).x(2,LP)];
end

F1 = @(z) (z.^(7/8)).*(-0.26849+0.797108*log(z))+0.45;
f1 = @(z) ((5/8)*((z./7).^(-1/8))).*(log(8) + (7/8)*log
   (z) - (7/8)*log(7) - log(0.72));
F2 = @(z) (z.^(7/8)).*(0.771872) - 0.6;
f2 = @(z) (5/8)*log(7/3)*((z./7).^(-1/8));
F3 = @(z) (z.^(7/8)).*(1.93388 - 0.797108*log(z))-1.8;
f3 = @(z) ((5/8)*((z./7).^(-1/8))).*(-log(8) - (7/8)*
   log(z) + (7/8)*log(7) + log(4.48));

figure(1)
h = cdfplot(Tc);
set(h, 'Color', 'k', 'LineStyle','--')
hold on;
hh = cdfplot(TT);
set(hh, 'Linestyle', '-.', 'Color', c('purple'), '
   LineWidth',1.6)
plot(z1, F1(z1), 'LineWidth', 1.2, 'Color',c('blue'))
hold on;
plot(z2, F2(z2), 'LineWidth', 1.2, 'Color',c('orange'))
hold on;
plot(z3, F3(z3), 'LineWidth', 1.2, 'Color',c('yellow'))

```

```
40 xlabel('Critical Surface Temperature')
41 ylabel('Probability')
42 title('')
43 legend('empirical CDF (MATCONT)', 'empirical CDF (
      analytical formula)', 'F_{Tc;1}','F_{Tc;2}','F_{Tc
      ;3}', 'Location', 'southeast')
44 grid off
45 ylim([0 1.05])
46
47 figure(2)
48 histogram(Tc, 'BinWidth', 0.1, 'Normalization', 'pdf',
      'FaceColor', 'none');
49 hold on;
50 histogram(TT, 'BinWidth', 0.1, 'Normalization', 'pdf',
      'FaceColor', 'none', ...
51      'LineStyle', ':', 'LineWidth', 1.6, 'DisplayStyle',
           'stairs', 'EdgeColor' ,c('purple'));
52 hold on
53 plot(z1, f1(z1), 'LineWidth', 1.2, 'Color',c('blue'))
54 hold on;
55 plot(z2, f2(z2), 'LineWidth', 1.2, 'Color',c('orange'))
56 hold on;
57 plot(z3, f3(z3), 'LineWidth', 1.2, 'Color',c('yellow'))
58 hold on;
59
60 xlim([0 4])
61 ylim([0 1])
62 xlabel('Critical Surface Temperature')
63 ylabel('Density of Probability')
64 legend('empirical data (MATCONT)','empirical data (
      analytical formula)','f_{Tc;1}','f_{Tc;2}','f_{Tc;3}
      ')
65
66 mean(Tc)
67 median(Tc)
68 skewness(Tc)
69 mean(TT)
70 median(TT)
71 skewness(TT)
```

Listing A.29: Probability density function and cumulative density function of the critical surface temperature.