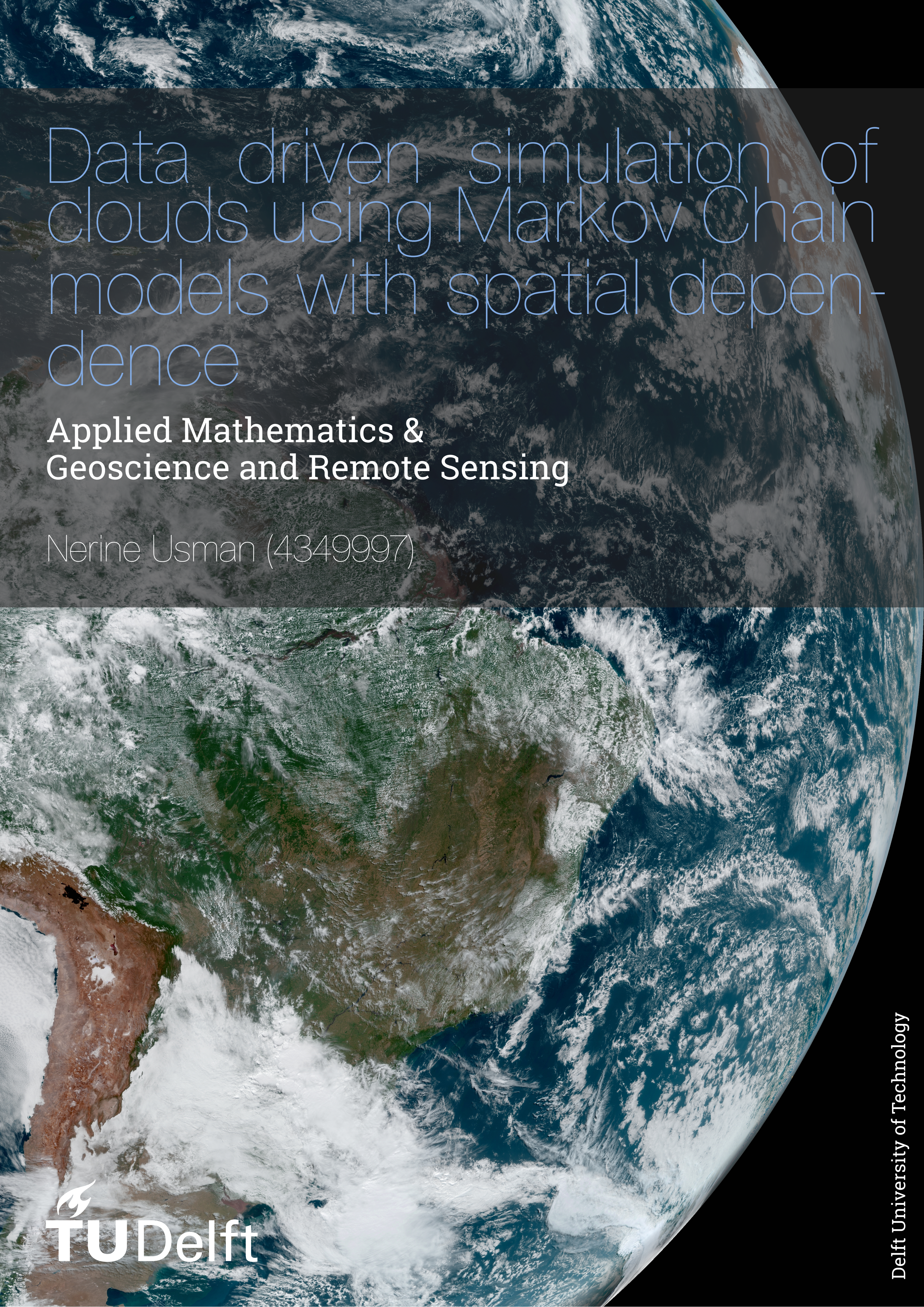


Data driven simulation of clouds using Markov Chain models with spatial dependence

Applied Mathematics & Geoscience and Remote Sensing

Nerine Usman (4349997)

Delft University of Technology

Data driven simulation of clouds using Markov Chain models with spatial dependence

Applied Mathematics & Geoscience and Remote Sensing

by

Nerine Usman (4349997)

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on August 30th

Student number:	4349997
Project duration:	Sep, 2021 - Aug, 2022
Thesis committee:	Prof.dr.ir. G. Jongbloed
	Prof.dr. A.P. Siebesma
	Dr.sc. F. Glassmeier
	Dr. C. Kraaikamp
	Dr. M.A. Schleiss

Abstract

Thanks to increasing computational resources, the grid sizes of global climate and weather prediction models are decreasing to scales at which subgrid processes, especially the evolution of clouds, can become spatially too variable for traditional deterministic parameterization approaches. We propose a stochastic parameterization in which clouds are represented by a Markov-Chain (MC) lattice model. The state of each lattice site is characterized by cloud top height (CTH) and cloud optical depth (COD). Based on these two parameters, clouds can be categorized into different cloud types (cumulus, stratus, cirrus, deep convection, etc.). In contrast to previous work, where the transition probabilities between different cloud types were calculated, we determine the transition probability densities in the continuous CTH-COD domain. We moreover explore the influence of neighbouring lattice sites on these transition densities to research whether it is possible to simulate cloud organization structures, without the use of extra external large scale variables or forcings. We estimate the parameters of the distributions using GOES-16 satellite images. Using these estimates, we perform simulations of the continuous state space Markov Chain over the lattice. This results in 2D spatial cloud evolutions, which spatial organization we compare to the original images. The simulations did not show any spatial structures as were observed in the original images. All the simulations resulted in a granulated image of different cloud types after only a couple of hours. A simulation from the MC model describing transitions from a finite number of cloud types converges to the observed occurrence frequencies, which is property of a MC. When describing the MC in a continuous state space the simulations still converged to the observed cloud state frequencies. However, introducing spatial dependence this property was not a given anymore. Since we were unable to reproduce the organizational structure from previous work using the GOES-16 satellite images, we cannot conclude whether it is possible to simulate cloud organization using the MC lattice model. A better understanding of these type of models was achieved. Neighbouring sides could diverge too fast in the performed models. We believe adding a stronger coupling between the cloud state evolution of neighbouring cells can improve the spatial models. Therefore one method to simulate correlated variables was tried to implement on the CTH evolution. This showed promising early results for further research. Also a better correction for horizontal advection on the original images could be performed to observe a specific cloud over time, which could improve the observed data quality.

Acknowledgments

This thesis is the final report for the two Masters I did, Applied Mathematics and Geoscience and Remote Sensing. It was interesting to be a bridge between two faculties which are literally facing one another. In this process I saw that it can be eyeopening for both faculties to discuss with one another. It would surprise I think how often research groups are trying to solve the same problems, only in a different context. We are often talking about the same, only giving it a different name, which makes it hard to see the connection.

Also, this thesis marks the end of a wonderful time for me at the TU Delft. There are some people that I would like to thank which made this journey possible and enlightened my life as a student. Since I am doing a project for two masters I was blessed with not one but three daily supervisors. First, I want to thank Geurt Jongbloed for the very insightful meetings. Every time I walked out of your door I felt more confident about how to continue my research. Not only your content-related remarks helped, but also when I felt a little bit down, you could help me by giving concrete tips on how to progress. Thank you. Pier Siebesma and Franziska Glassmeier supported me on behalf of the GRS master on the atmospheric side of this thesis. Our different backgrounds resulted in very interesting discussions about cloud organization and everything around the field. Thank you both. I want all three of you for being open to the collaboration with another faculty. I hope it brought you new insights too. Also I would like to thank Marc Schleiss and Cor Kraaikamp for being in the assessment committee as well. I want to thank Niels Janssen for helping me getting started working on the VRLab. You gave me very clear and kind instructions on how to get access and run codes, which has saved me so much time in the rest of the project, and you were always available if I had a small question.

Next, I would like to thank all my study buddies. It was a weird time with a lock down during a large part of my project and it has helped so much to study together with you. I would like to thank Emmy and Menno especially, who have been there since the Bachelor. I also studied a lot together with people from Krashna. I want to thank all the Krashnese for a warm home during my whole study period. Krashna has become a really special place for me. Everyone is so kind, open to all kinds of people, funny and respectful. Thank you all!

Finally, there are those who are incredibly dear to me. Dad, Mum, Ismene, Marit, Bregje and Redmer, I want to thank you so much for always being there for me, asking me the most relevant questions and making me feel so close to you.

Nerine M.E. Usman
August, 2022

Contents

1

Introduction

Numerical global climate and weather models are extensively used to get a better understanding of the system Earth. Thanks to increasing computational resources the grid sizes of these models have decreased, resolving more processes at smaller scales. However, in general there are phenomena which happen at scales smaller than the grid spacing size. Parameterizations are needed to model the influence these subgrid scale (SGS) processes have on the resolved processes. These parameterizations can be inferred from small scale models which resolve the SGS processes and high resolution observations.

It is agreed that a major source for uncertainty in global climate models comes from the modelling of clouds. Clouds have both a cooling and a warming effect on the Earth. Solar radiation is reflected which causes cooling, but at the same time long wave radiation is trapped underneath the clouds which causes warming. In between different climate models a large spread is observed in the estimation of the net result of clouds on the Earth temperature (Forster et al., 2021). This is mainly due to the effect that the processes which influence the development of clouds, such as convection and boundary layer processes, are SGS processes. So therefore these processes cannot be completely resolved, but have to be implemented in the models using parameterizations.

For the parameterization of convection in clouds, the global model grid sizes are becoming so small; they come close to the so called Gray Zone. In this zone the small scale convective updrafts are not completely resolved yet, but their scale is only slightly smaller than the grid sizes, which causes high variability at the grid as illustrated in Figure 1.1. Because of the high variability the convection cannot be parameterized by a deterministic function, so therefore stochastic parameterizations for convection are introduced (e.g. Alexandrov and Marshak, 2017; Bengtsson et al., 2013; Dorrestijn et al., 2013b; Hagos et al., 2018; Khouider et al., 2010; Prigarin and Marshak, 2009; Sakradzija et al., 2016).

There are many different methods to introduce a stochastic element in the parameterization of cloud convection. One of the methods proposed is the use of Markov Chain (MC) lattice models, introduced by Khouider et al. (2010). These MC models are defined on a 2D horizontal grid, where every column of atmosphere is classified from a finite number of cloud types. The MC defines the probabilities p_{sk} on a column with cloud type s transitioning to cloud type k in the next time step. Dorrestijn

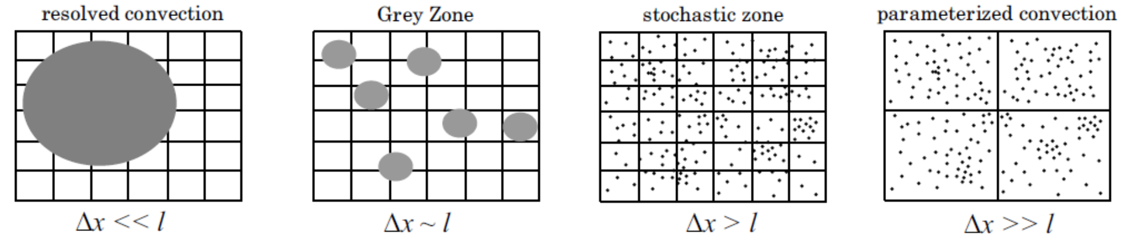


Figure 1.1: Illustration of the grid size Δx and the length scale l of a convective updraft, which are depicted by the grey dots. For high-resolution models ($\Delta x << l$) the convection is explicitly resolved. For very coarse models ($\Delta x >> l$ the effect of the convection can be parameterized, using for example mean values, which would give a very good description about the state in the cell. When the grid size and the process scale are close to one another there is a high variability in number of convective updrafts per cell among the grid. In this case convection can still be parameterized but a stochastic element is needed to capture the high variability. Figure adapted from Dorrestijn (2016, Fig. 1.9)

et al. (2013a) continued on this work, estimating the transition probabilities from one to another cloud type using Large Eddy Simulation (LES) data. LES simulate convection with very high resolution, giving a fully resolved convection data set. Here it was remarked that the transition probabilities are not constant, but can vary based on additional external variables, such as e.g. temperature, relative humidity or Convective Available Potential Energy (CAPE), and the cloud types in the neighbouring columns. CAPE is a measure which is often used as an indicator for the possibility of thunder storm development. The transition probabilities are conditioned on additional external variables available in the larger scale grid and/or on a function of the cloud types in adjacent atmosphere columns. In Dorrestijn et al. (2015) and Dorrestijn et al. (2016) these MC models were implemented in global climate models, but with the transition probabilities only conditioned on additional external large scale variables, and neglecting the influence of the cloud types at neighbouring cells. These models were trained based on observational radar data for the transitions and additional data for the external variables in Darwin, Australia. In this way the evolution of each lattice site is only indirectly coupled to the evolution at a neighbouring site through the macroscopic variables.

In this study we use the models from Dorrestijn et al. (2013a) as a starting point. In this paper four different MC models were discussed; (1) the MC with constant transition probabilities, including, (2) including the influence of external additional variables on the transition probabilites, (3) including the influence of the state of neighbouring sites and (4) a combination of both influences. In this this we will neglect the influence of external additional variables on the transition probabilities. We are interested whether it is possible to simulate the spatial cloud organization if the transition probabilities are only conditioned based on information from the neighbourhood of the lattice site. So in this thesis we do not condition on other large scale variables which influence the evolution of the cloud types. If the models are able to capture the spatial behaviour of the clouds they could form an interesting basis for the construction of parameterizations, since simulating the MC models is computationally inexpensive.

We question the use of a finite number of cloud types as was done in previous research, as we expect the transition probabilities to vary based on the exact cloud state. Therefore we will define the MC models such that the state of the cloud is described in a continuous domain, instead of a domain consisting of a finite number of cloud types. We will train the models in this study based on GOES-16 satellite imagery. This satellite provides relatively high resolution images, both spatially and temporally, for the Cloud Top Height (CTH) and Cloud Optical Depth (COD). The latter variable is a measure that describes the 'thickness' of a cloud. Based on these two variables the state of a cloud can be described and using the International Satellite Cloud Climatology Project (ISCCP) classification a cloud can be classified to a certain cloud type, e.g. cumulus, cirrus, or stratus clouds. We will use these two variables will span the continuous domain in which we determine the cloud state.

The report is structured as follows. In Chapter 2 we will define the mathematical framework of our MC lattice model. In Chapter 3 we examine the available data from the GOES-16 satellite and how we need to adjust it to be able to use it for training of our models. Additionally, we define metrics to quantify the spatial cloud organization from our simulations. In Chapter 4 we will define and clarify the different models in increasing complexity. As a reference we start with implementing a simple MC, followed by two extended versions of this reference model. In the first extension we include conditioning on cloud type state in the neighbouring lattices, which makes the model a *stochastic cellular automaton* (SCA). In the second extension we change the state space of the sites to the continuous COD/CTH space. Finally we consider also a version where we combine the two extensions. In Chapter 5 we will show the results of the simulations of the different models and we will discuss them in the last Chapter 6. In this last chapter we will also conclude our findings.

2

Modelling cloud evolution with a Markov Chain

We are interested whether it is possible to model the spatio-temporal evolution of cloud clusters based on the current state of the cloud and its spatial environment. In the models the state will be described purely by the height of the cloud top and the cloud optical depth, since these variables are provided by the GOES16 satellite and give a good bases for cloud type classification. This latter variable is a measure related to the 'thickness' of a cloud. In previous work the states of the clouds were categorized into a finite number of cloud types, based on similar variables (e.g. Dorrestijn et al., 2013a; Khouider et al., 2010). The probability that a cloud transitions from one to another cloud type was observed from high resolution simulation or observation data. The evolution of a cloud can then be modelled by a Markov Chain (MC). In this chapter we will define the general structure of our models which is based on the these previous works.

Markov Chains are commonly used to describe many different random processes. A simple Markov Chain is a sequence of random variables X_t, with $t = 0, 1, 2, \ldots$, of which each can take a finite number of states and the probabilities to transition from one state to another are given. Say for example we consider a MC with four possible states in $\Omega = 1, 2, 3, 4$. The probability to transition from s to k is called a transition probability and is denoted by p_{sk} for $s, k \in \Omega$. Assume the transition probabilities are given by the arrows in Figure 2.1. So, if we have for example $X_4 = 2$ at time $t = 4$, then for X_5 we know that $\mathbb{P}(X_5 = 1 | X_4 = 2) = p_{21} = \frac{1}{2}$, $\mathbb{P}(X_5 = 2 | X_4 = 2) = p_{22} = \frac{1}{4}$, etc.. One of the main properties for Markov chains is that knowing any extra information about previous states, does not change the probability distribution for the states in the future. So knowing that $X_3 = 1$, does not change our knowledge about the distribution of X_5 if we already know the value of X_4. Therefore, Markov Chains are said to be 'memoryless'.

In the MC models for the cloud types, we will use different cloud types as states. For example 1 corresponds to clear sky (not really a 'cloud type' but here we will use this terminology anyway), 2 to cumulus, 3 to stratus and 4 to cumulonimbus. Once we know all the transition probabilities from one to another cloud type, a simulation of the cloud type evolution can easily be done by drawing each next state randomly according the the corresponding transition probabilities.

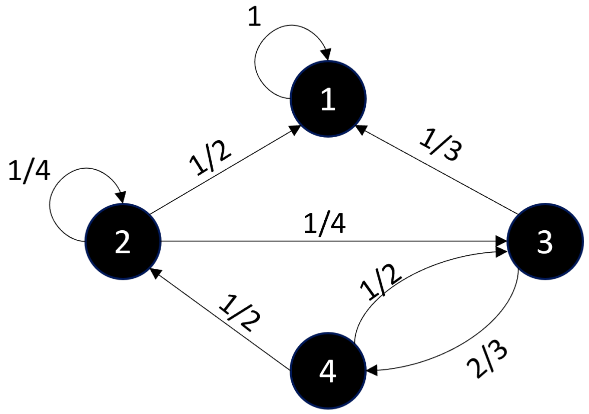


Figure 2.1: Example of Markov Chain Model. The states are depicted by the dots, and the transition probabilities by the arrows and values.

In the proposed models we will extent this simple MC by defining it on a continuous state space domain, instead of the finite number of states. This requires some different mathematical notation which will be explained below.

In this chapter we will define the general structure of the models, of which different versions will be presented in Chapter 4. We will start with defining the key variables for the continuous state space Markov Chain in Section 2.1, and introduce a general formulation for the transition probability density in Section 2.2. After this setup, we can formulate our observational data in the same setting (Chapter 3) and apply models with different choices for some of the functions defining the model (Chapter 4).

2.1 Markov Chain in continuous state space

In previous work for stochastic parameterization a central element was the discretization of the microscopic scale states (e.g. Dorrestijn et al., 2013a; Khouider et al., 2010), which were then modelled as a Markov Chain (MC). Here we keep the MC approach, but extend the state space to a mixture of a continuous space for the cloud state and one discrete state for the presence of clear sky. We use Cloud Top Height (CTH) and Cloud Optical Depth (COD) to determine the state of a cloud, since both these values are available from the GOES-16 data as we will see in Chapter 3. Let us denote the cloud state

$$S_i(t) = (D_i(t), H_i(t)) \in \Omega,$$

at time t and location i, with D the logarithm of COD, H the CTH and $\Omega = \Omega_c \cup \{(0,-1)\}$ the state space, with Ω_c the state space for presence of clouds. Since clouds are obviously always above sea level, the CTH can only take positive values, which gives $\Omega_c := \mathbb{R} \times \mathbb{R}^+$. However, in the absence of clouds, the COD and CTH are not defined, so therefore we assign the state $(D, H) = (0, -1)$ to locations with no clouds. We come to the specific state space domain which is a union of a continuous and discrete domain. See Figure 2.2 for an illustration of the state space Ω. We use the logarithm of COD because of the range of observations we will see in section 3.1.

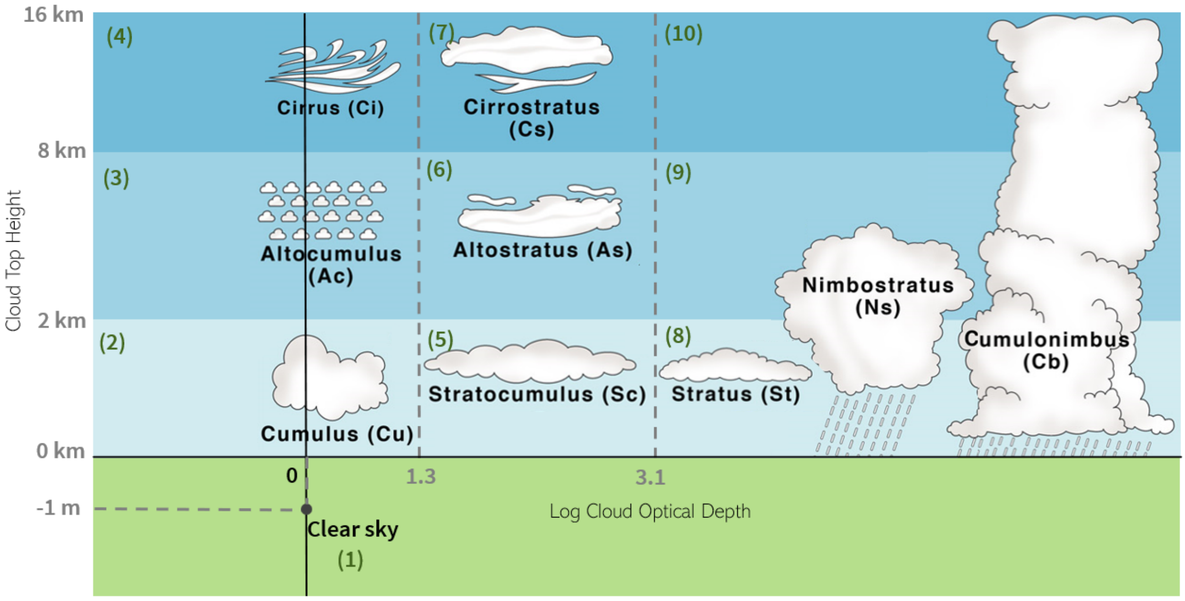


Figure 2.2: The state space Ω of $S_i(t)$. The state for clear sky is at the virtual point (0, -1) and the clouds have a state $(d, h) \in \mathbb{R} \times \mathbb{R}^+$, with d the log of COD and h the CTH. Based on these values the International Satellite Cloud Climatology Project (ISCCP) Cloud Classification scheme classifies different cloud types. The corresponding numbers for each cloud type are given within parenthesis in dark green. Figure inspired by Ormes (2017), Fig 4.

For understanding of the data and comparison with the previous model described by Dorrestijn et al. (2013a) we use the International Satellite Cloud Climatology Project (ISCCP) Cloud Classification to classify the cloud states. This scheme links CTH and COD to a cloud type as in figure 2.2. This gives us 10 cloud types, 9 for the clouds and one for the clear sky state. Let the function c map a state to a certain cloud type,

$$c : \Omega \to \{1, \dots, N\}, \tag{2.1}$$

with $N = 10$ the number of cloud types. Each number corresponds to one of the cloud types in Figure 2.2, (1) clear sky, (2) cumulus, (3) altocumulus, (4) cirrus, (5) stratocumulus, (6) altostratus, (7) cirrostratus, (8) stratus, (9) nimbostratus, (10) cumulonimbus (which can also be referred to by *deep convection*). So for example $c(0, -1) = 1$ is the classification for clear sky locations, and $c(4, 5000) = 9$, corresponding to Nimbostratus.

The time evolution of $S_i(t)$ is modelled by a Markov Chain as in Dorrestijn et al. (2013a). This means that we look at the evolution of $S_i(t)$ at discrete times $t = 0, \Delta t, 2\Delta t, \dots$ for a fixed time step $\Delta t > 0$. Contrary to Dorrestijn et al. (2013a) we have a continuous state space instead of a state space consisting of a finite number of discrete cloud types. Therefore we cannot simply write the transition probabilities from one to another state but here we denote, the transition probability from state s to a subset of states A

$$p(s, A) = P(S_i(t + \Delta t) \in A | S_i(t) = s), \tag{2.2}$$

for any $s \in \Omega$ and Borel set $A \subset \Omega$. We assume that the Markov Chain is time and location invariant, meaning that the transition probabilities do not depend on time t or location i. Therefore we drop the dependence on both these variables in the notation for the left hand side of Eq. (2.2). The validity of this assumption will be investigated in Section 3.1 based on the training data. In this representation the state at location i for the next time step depends only on the current cloud state at this location. As Dorrestijn et al. (2013a) also already noticed, this transition is not only dependent on the current state but is influenced by other macroscopic variables $X_i(t)$, e.g. convective available potential energy (CAPE), or relative humidity (RH). Therefore the transition probabilities are also assumed to depend on, a subset of, these macroscopic variables. The transition probabilities then become

$$p_x(s, A) = P(S_i(t + \Delta t) \in A | S_i(t) = s, X_i(t) = x), \tag{2.3}$$

which makes the MC a *conditional Markov chain.*

Through the conditioning on a macroscopic variable there is an indirect relationship between two adjacent points, since the macroscopic variables will be strongly correlated. Therefore the transition probabilities for the states at these points will be very similar.

Here, we do not consider conditioning on macroscopic variables but we focus on only a direct coupling between adjacent variables. By isolating only the local interaction, we are interested in whether it is still possible to generate the (spatial) variability and organization in clouds as observed. By conditioning on the current state and the state of the neighbouring cells, the Markov Chain becomes a *stochastic cellular automaton (SCA)* with transition probabilities

$$p_\gamma(s, A) = P(S_i(t + \Delta t) \in A | S_i(t) = s, g(S_{\{i\}}(t)) = \gamma), \tag{2.4}$$

where $S_{\{i\}}(t)$ are the states $S_j(t)$ of locations j in the neighbourhood $\{i\}$ of location i, e.g. the 8 grid points around location i, g a neighborhood state aggregation function, and $\gamma \in \mathbb{R}^m$ the realization of the aggregated state of the neighbourhood. If we would condition on all the states in the neighbourhood, depending on the size of the neighbourhood, the number of extra variables we condition on increases quickly. The states in the neighbourhood are probably strongly coupled as well. It might for example not matter which of the neighbours contain which state, but only how many are a certain cloud type. Therefore we condition on a selection of m features from the states in the neighbourhood, using a function g. The choice for g will be explored later on in section 4.4, as it should capture the relevant information about the neighbourhood that influences the transition probabilities. But for now it is only important to realize that γ contains m values, with $m \in \mathbb{N}$, which all contain information about the cloud state of the neighbourhood. For a simple model m is preferably small.

Remark that in this thesis we will simulate the states of $S_i(t + \Delta t)$ and $S_j(t + \Delta t)$ for two neighbouring locations i and j independently. The two distributions from which we draw the two realizations will be similar since $S_i(t)$ and $S_j(t)$ will be most likely very similar and the cloud states in the neighbourhoods as well, but the realizations are independently simulated.

2.2 Transition density function

The continuous analogue of a transition probability p_{sk} from discrete state s to k is the *transition density* $p_\gamma(s,k)$ with respect to a measure ν such that

$$p(s,A)=\int_A p_\gamma(s,k)d\nu(k). \tag{2.5}$$

This transition density is the probability density function of $S_i(t+\Delta t)$ given $S_i(t)=s$ and $g\left(S_{\{i\}}(t)\right)=\gamma$. Due to the special state space Ω which is a union of a continuous and a discrete domain, this transition density has a mixture representation. The transition density is

$$p_\gamma(s,k)=p_{s,\gamma}\mathbb{1}_{\{(0,-1)\}}(k)+(1-p_{s,\gamma})f_\gamma^{(c)}(k|s)\mathbb{1}_{\mathbb{R}\times\mathbb{R}^+}(k), \tag{2.6}$$

with respect to dominating measure ν which assigns mass $\nu(A)=\mathbb{1}_A((0,-1))+\int_A d\lambda(x)$ for $A\subset\Omega$, where λ denotes Lebesgue measure on $\mathbb{R}^2$, $p_{s,\gamma}$ the probability on transitioning to a clear sky state, $f_\gamma^{(c)}(k|s)$ the *cloud transition density function*, which will be explained in the text below, and $\mathbb{1}$ the indicator function, defined as

$$\mathbb{1}_A(x)=\begin{cases}1 & \text{if } x\in A,\\ 0 & \text{if } x\notin A.\end{cases} \tag{2.7}$$

This transition density requires some interpretation. Say for example we have state $S_i(t)=s$ and a neighbourhood state summarized by $g(S_{\{i\}}(t))=\gamma$ at time t. The probability that the next state $S_i(t+\Delta t)=k$ is a clear sky, i.e. $k=(0,-1)$, is $p_{s,\gamma}$, since then the first indicator function in Eq. (2.6) will become one and the second zero. The probability $p_{s,\gamma}$ is a function of the current state of the pixel and the states in the neighbourhood. The transition to a cloud for the next time step, happens with probability $1-p_{s,\gamma}$. In that case $k\in\mathbb{R}\times\mathbb{R}^+$ so we end up in the second term of the transition density. Here the cloud transition density $f_\gamma^{(c)}(k|s)$ denotes the pdf of the cloud state given γ and s, where the superscript c highlights that this is the transition density to a *cloud* state. This distribution function is defined such that the Lebesgue integral over the state domain for clouds equals one, i.e.

$$\int_{\mathbb{R}\times\mathbb{R}^+} f_\gamma^{(c)}(k|s)d\lambda(k)=1,$$

for any $s\in\Omega$ and $\gamma\in\mathbb{R}^m$.

The probability to evolve from state s within a subset of states A equals

$$p_\gamma(s,A)=p_{s,\gamma}\mathbb{1}_A((0,-1))+(1-p_{s,\gamma})\int_A f_{s,\gamma}^{(c)}(k|s)d\lambda(k), \tag{2.8}$$

for any Borel set $A\subset\Omega$. Note that indeed the next state is within the state space Ω with probability 1:

$$\begin{aligned}p_\gamma(s,\Omega)&=p_{s,\gamma}\mathbb{1}_\Omega((0,-1))+(1-p_{s,\gamma})\int_\Omega f_\gamma^{(c)}(k|s)d\lambda(k) &(2.9)\\ &=p_{s,\gamma}+(1-p_{s,\gamma})=1. &(2.10)\end{aligned}$$

In chapter 4 we will show models with different examples for the two dimensional cloud transition density function $f^{(c)}$ and for the choice of the aggregating function g (Eq. (2.4)), e.g. what spatial information we consider important for the cloud evolution. But first we turn to our available data and how we will measure the spatial variability and organization in the following chapter.

3
A view at cloud satellite images and cloud organization

In this chapter we will have a view at the data on which we will train the model. We use imagery from the GOES-16 satellites at which we will describe in Section 3.1. In Section 3.2 cloud organization is discussed and how we can quantify it. The metrics which we will later use to determine the performance of our models are defined in this section.

3.1 Introducing GOES-16 satellite images

This study focuses on a near-equator area above the pacific ocean (5°S to $15°N$, 50°W to 30°W), at the locations at least 100 km out of the coast of Brazil in the months December, January and February (DJF), see Figure 3.1. During this period hurricanes are not occurring.

The area of interest is chosen to be over sea such that the observations are as much time and location invariant as possible. Surface heating due to sunlight has a large influence on the daily development of clouds. Due to the high specific heat capacity of water, the ocean warms much slower and therefore this diurnal effect is much less present over ocean than over land. Also over land there is a lot of variability in soil, moisture, vegetation etc., which could all influence the evolution of the clouds in the air. Taking the study area over sea, makes the surface much more homogeneous.

The Geostationary Operational Environmental Satellite (GOES)-16, also known as GOES-East is operated by the National Aeronautics and Space Administration (NASA) and the National Oceanic and Atmospheric Administration (NOAA). Using its Advanced Baseline Imager (ABI), it provides 16 spectral bands with high spatial and temporal resolution. The 16 channels include two visible channels, four near-infrared channels and ten infrared channels. The ABI takes images at three different spatial scales; Mesoscale, Contiguous United States (CONUS) and the Full Disk. All taken at different temporal resolutions, which depend on the scanning mode the satellite is configured to. The mesoscale images cover the smallest extent but with the highest temporal resolution and are mainly used to track specific events, such as hurricanes propagating towards the Americas. The CONUS covers North America, and the Full Disk the whole earth

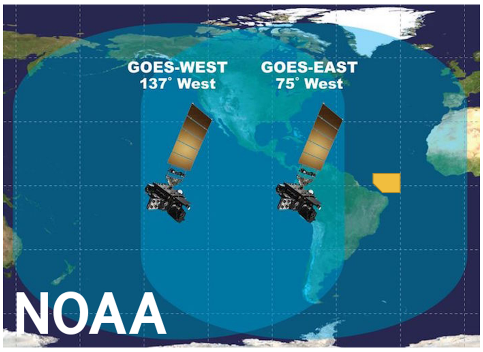


Figure 3.1: An overview of the extent of the full disc images of the GOES-16 (GOES-East) satellite. The yellow area indicates the study area.

observed from the satellite. So the only images containing our area of interest are the Full Disk images. Since April 2019 these are taken every 10 minutes.

For our model we obviously use the Cloud Top Height and Cloud Optical Depth product from the ABI but we will also use the so called Derived Motion Winds (DMW) product. For the training of our model we will need a time sequence of states of a cloud, but due to horizontal wind, the location of the cloud will change in subsequent images. Therefore the DMW will be used to determine the location of a cloud from one to another subsequent image. In the following subsections these different products are described.

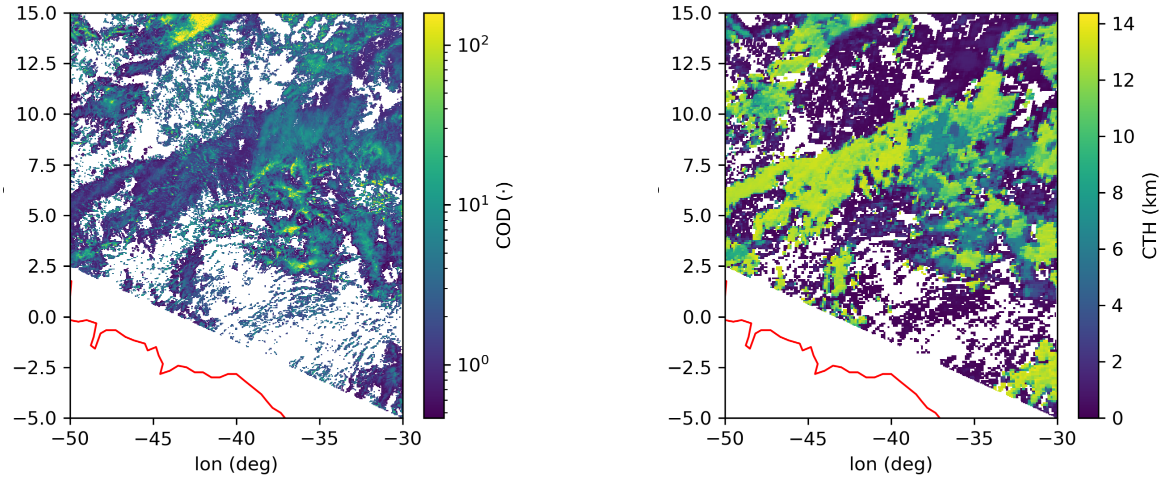
3.1.1 Cloud optical depth

The cloud optical depth (COD) is a measure related to the thickness of a cloud. It represents in some way how much water there is contained in a vertical column of a cloud. The COD during day light is calculated using band 2 (0.64 μm, red visible) and band 6 (2.25 μm, near IR) during presence of day light (Walther et al., 2013). There is also an algorithm for night time available, but the performance is much worse giving often much lower values for the COD (Minnis and Heck, 2012). So in this study we will only use daytime observations. For the daytime algorithm, daytime is defined if during that time the solar zenith angle is smaller than 65°. This constraint decreases the amount of good quality images substantially. As a back-of-the-envelop estimation, neglecting the Earth's axial tilt, this leads to only

$$\frac{2 \cdot 65^\circ}{360^\circ} \cdot 24 \text{ hours} \approx 8.7 \text{ hours}$$

of daytime images per day at the equator. Also the COD is only calculated for pixels that are classified to be cloudy or probably cloudy.

The COD represents the vertical optical thickness between the top and bottom of an atmospheric column. It has no unit and is calculated from the two bands by the Daytime Cloud Optical and Microphysical Properties algorithm (DCOMP) provided by the NOAA (Walther et al., 2013). This algorithm also calculates the Cloud Particle Size (CPS). From the CPS and the COD an estimate for the Liquid Water Path (LWP) and



(a) **(b)**

Figure 3.2: (a) The COD and (b) CTH for the study area at 2020-12-10 16:04 UTC. The red line indicates the coast of Brazil. The lower triangle has been removed from the area of interest such that all our observations are more than approximately 100 km out of the coast.

Ice Water Path (IWP) are derived. The final product for COD has a spatial resolution of 4km at nadir. In Figure 3.2a the COD for one image at the study area is shown.

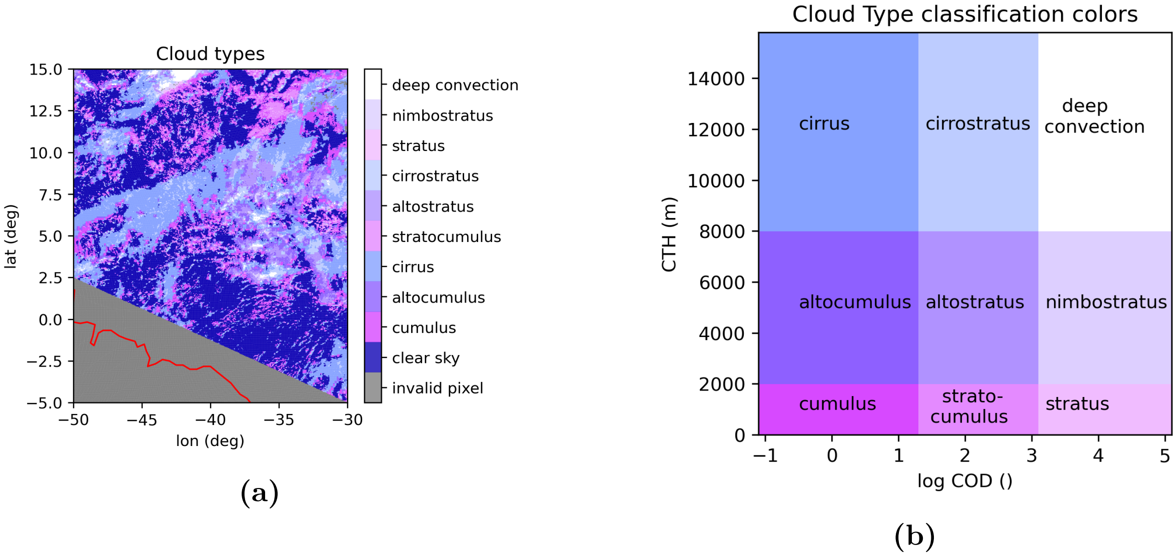
3.1.2 Cloud top height

The other essential variable for the cloud type determination is the Cloud Top Height. This variable is generated by the ABI Cloud Height Algorithm (ACHA) together with the cloud-top temperature, cloud-top pressure and cloud layer, Heidinger (2012). This algorithm only uses infrared observations and can therefore generate images all day, but due to our limitation in COD we will only use the day time images.

Unfortunately due to the lower spatial resolution of the infrared bands the cloud top height is only available at a spatial resolution of about 10 km at nadir. In Figure 3.2b the CTH is shown for the same scene as the COD in Figure 3.2. It can be seen that the structures are slightly less detailed.

Based on the COD and CTH we can determine the cloud types for the particular scene in Figure 3.2 according to the ISCCP classification from figure 2.2. This results in Figure 3.3.

Using all images from the 2020-2021 (DJF) the histograms of COD and CTH are as in Figure 3.4. Here it can be seen that there is a clear bi modality in the CTH. Most of the clouds are low clouds (below 2 km) or high clouds (11-14 km). Clouds above 15 km are very unlikely and only occur for higher COD. The logarithm of the COD is used to show this histogram, otherwise it is not possible to distinguish between the smaller values. It can be seen that there is an isolated peak at logCOD = 5. This is due to the measurement range of the ABI, all higher values are mapped to this threshold. Also



(a)

(b)

Figure 3.3: (a) The ISCCP classification for based on the images from Figure 3.2. The color coding which is based on the COD and CTH is indicated in (b).

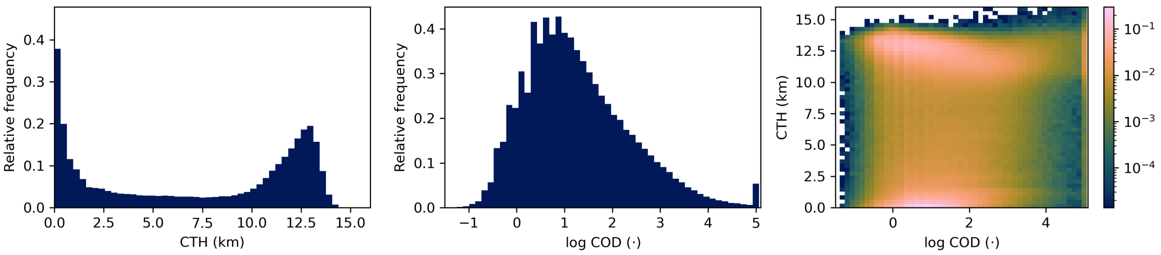


Figure 3.4: Normalized istograms of the (left panel) CTH, (middle panel) COD and (right panel) both products together of all the data in DJF (2020 - 2021). The histograms are normalized such that the integrals sum to one.

there are several peaks for the smaller values due to the small differences in the values caused by the log transformation, which the instrument can not distinct.

3.1.3 Derived motion winds

In our models we want to simulate the temporal evolution of clouds. If we would observe the cloud states at fixed locations over time, we would not observe the evolution of a cloud. Due to the presence of advection, we would observe a sequence of clouds passing by one location. The difference between the states of two subsequent observations would not only contain a change in time but also spatially. Therefore we have to determine the spatial path a cloud travels over time. The states at this path then form the sequence of states which are a realization of the MC we try to model.

Suppose we observe the state $s_{i_0}(t = t_0) = s$ for a cloud at time $t = t_0$ and a location i. We have to determine the location sequence $i(t_i)$ of the location of the cloud in the following time steps with $t_j = t_0 + j \cdot \Delta t$, $j = 0, 1, 2, \ldots$, $\Delta t > 0$ and $i(t_0) = i_0$. Then we take

$$i(t_{j+1}) = i(t_j) + \Delta t \cdot \mathbf{u}(i(t_j), t_i), \tag{3.1}$$

where $\mathbf{u}(i,t)$ denotes the horizontal wind velocity at location i and time t. Using this location sequence we can find the subsequent states observed for the cloud which are $s_{i(t_j)}(t_j)$, with $j = 1, 2, \ldots$. Later in this report, we will always drop the dependency on time for the location, assuming that all observations are already corrected for advection.

There is a level 2 product provided by the ABI which generates an estimate for the horizontal wind velocity field $\mathbf{u}$ called Derived Motion Winds (DMW), (Daniels et al., 2019). The product generates derived motions for six different spectral bands separately. For each of the bands it uses three subsequent images to determine wind velocities. It performs an initial feature targeting on the middle image, and searches then for the same features in the other two images. Based on the displacement in the images an estimate for the velocity at the target location is made. Depending on the spectral band, different features can be observed at different locations in the area. Combining the motion estimates for the features at all the spectral bands improves the spatial coverage of motion estimates. This results in an unregulated grid of observations, contrary to the COD and CTH product. We use nearest neighbours to determine an estimate for the horizontal wind velocity at any location in the area. An illustration of the determination of the spatial path of a cloud can be found in the results section 5.1.2.

Unfortunately the Derived Motions are generated only every 60 minutes, giving a much lower temporal resolution than the images of COD and CTH, which are available every 10 minutes.

3.1.4 Data adjustments and preparations to fit the models

To be able to use the above described products for the training of our models we need to adjust the raw images. This is to obtain measurements of COD and CTH at fixed locations at an equidistant grid, in case there was a cloud present, including the location of the observed state in the next time step. Also the three products (COD, CTH and DMW) at the Full Disk images for the three months (DJF) of one year combined together consist of 192 GB of data. After step 5 of the list below, we have reduced this amount to about 5 GB. Taking only specific time series of clouds in the last step this amount is reduced even more.

Therefore we need to

1. reduce the Full Disk images to our study area,
2. interpolate the CTH observations to the same locations as the COD observations,
3. remove all the observations where the COD or CTH is flagged as being a bad quality,
4. project the locations of the observations to a coordinate system in km instead of degrees,
5. determine the location of the cloud for each pixel in the next time step. This is done by determining the horizontal wind velocity $\mathbf{u}$ and applying equation (3.1).

The horizontal wind velocity is determined using a nearest neighbours estimate from the DMW product, since this product is provided on an irregular grid,

6. make a time series of the cloud types starting from a limited number of pixels. For most of the models we determined the time series of pixels which were 15 km apart. This was done, since neighbouring pixels are very highly correlated and contain therefore almost the same information.

3.2 Cloud organization metrics

Our main interest is whether our simulations which only uses local rules can capture the spatial structure and variability of clouds. For this to verify, we need to agree on metrics to quantify the spatial structure.

In the past many metrics have been used to measure spatial structures of clouds. Janssens et al. (2021) investigated 21 of the commonly used metrics and showed using principal components that with only four derived metrics (the first four principal components) the cloud pattern can be described. The four dimensions are aligning with the following characteristics of the cloud field; the characteristic length, void size, directional alignment and horizontal cloud top height variance.

We will only use the first two principal components. Many of the original 21 metrics are spanned by these two. There are multiple pairs of metrics that would span almost the same space as these first two principal components. So a choice can be made for two of the original metrics to quantify the cloud organization pattern. Some metrics are easier to implement than others so in this study we will use the cloud fraction, which is very straightforward to compute and the Organization Index I_{org} which was first introduced by Weger et al. (1992).

Cloud type filling fraction

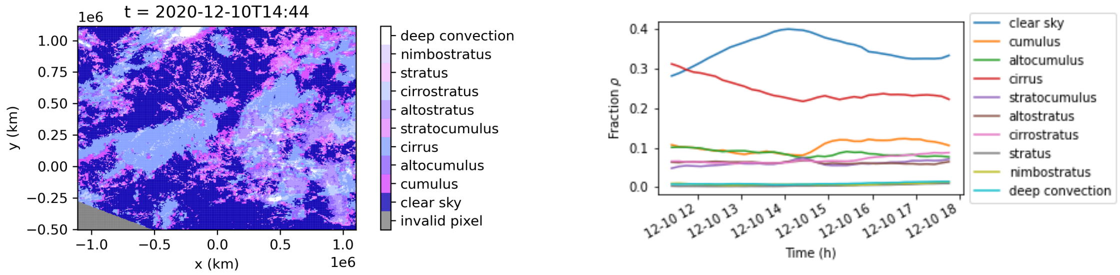
We define the cloud type filling fraction at time t,

$$\rho_\alpha(t) := n^{-1} \sum_{i=1}^{n} \mathbb{1}\left(c(S_i(t)) = \alpha\right), \tag{3.2}$$

for $\alpha \in \{1, \ldots, N\}$ a cloud type where c is as defined in Eq. (2.1) and n is the number of locations which we observe. The cloud fraction τ is then easily calculated from the clear sky ($\alpha = 1$) fraction,

$$\tau(t) = 1 - \rho_1(t). \tag{3.3}$$

In Dorrestijn et al. (2013a) instead of the below described organization index I_{org} the standard deviation of the cloud type filling fractions $\sigma_\alpha(t)$ over macroscopic blocks was used as a measure of spatial variability. This metric is easier to calculate than the I_{org} and less sensitive to cloud object identification. Therefore we will use this metric as a first description of the spatial structures. When this metric gives promising results



(a) **(b)**

Figure 3.5: (a) One scene from the observations which we will use as starting state for the simulations using the same color code as before, and (b) cloud type fractions ρ (solid) for this scene and the images after this time stamp for 6 hours. Only the seven most present cloud types are shown.

I_{org} can be calculated for a more exact analysis. For the calculation of $\sigma_\alpha(t)$ we divide the grid in N macroscopic blocks which contain $n = l \times b$ locations, where l and b are the number of locations in the north-south and east-west direction respectively. Then

$$\sigma_\alpha(t) = \frac{1}{N}\sum_{j=1}^{N}(\rho_\alpha^j(t) - \rho_\alpha(t))^2, \tag{3.4}$$

where $\rho_\alpha^j(t)$ is the filling fraction of cloud type α in the j-th macroscopic block and $\rho_\alpha(t)$ the mean filling fraction over the whole area. This standard deviation is very dependent on the choice of N. In practice we will try a few different choices of N, but one has to remind that only for the same value of N the standard deviations can be compared.

In figure 3.5 the cloud type fractions are shown for a sequence of images of one day.

Organization Index I_{org}

The Organization Index I_{org} was first described by Weger et al. (1992). It is now commonly used to determine cloud organization structure. The index compares the cumulative distribution function of the Nearest Neighbour distances (NNCDF) with a Weibull distribution. For each cloud the distance to its nearest neighbour is computed. The Weibull distribution is expected when all clouds are randomly distributed over the field. In Appendix A of Tompkins and Semie (2017) a nice illustration of the computation of I_{org} based on this comparison is shown. If the clouds are randomly distributed over the field, $I_{org} = 0.5$. If the clouds are more clustered (regular) distributed the value increases (decreases), with 0 and 1 as minimal and maximum value.

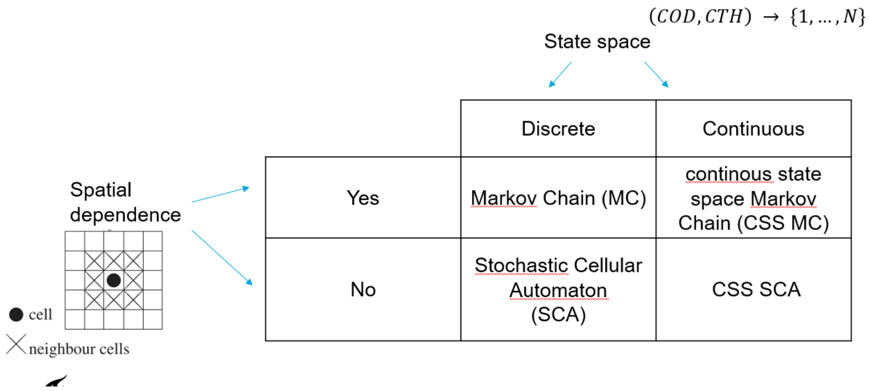
For the calculation of the Organization Index we need to determine the different clouds as objects, to be able to calculate the nearest neighbour distances. Inspired by Böing (2016) a paint-bucket fill algorithm could be used to determine distinct clouds. This algorithm makes sure that clouds connected through very elongated objects are not classified as one cloud.

4

Models

From the previous two chapters we obtain the model formulation and an insight in the available images provided by GOES16. In the model formulation in Chapter 2 we left flexibility in the choice for possible cloud transition density functions $f^{(c)}$ and the neighbourhood aggregation function g. Different choices lead to different models with varying complexity, which will be discussed in this chapter.

An overview of the four different models which we will discuss is shown in Figure 4.1. In the first two sections of this chapter we discuss the models described in Dorrestijn et al. (2013a) without conditioning on external factors, trained on the GOES16 satellite data. The first one (Section 4.1) is the simplest Markov Chain (MC), which has a discrete state space and no spatial dependence. As extension of this model we also implement the Stochastic Cellular Automaton (SCA), which contains spatial dependence. This model will be a reference of our later model in which we also include spatial dependence. In Section 4.3 we extend the original simple MC by expanding the state space to a continuous state space for clouds, but keep don not include the spatial dependencies yet. We will refer to this model by continuous state space MC (CSS MC). Here we search for a suiting cloud transition density function $f^{(c)}$ (Eq.(2.6)) and check how well it fits



	Discrete	Continuous
Yes	Markov Chain (MC)	continous state space Markov Chain (CSS MC)
No	Stochastic Cellular Automaton (SCA)	CSS SCA

Figure 4.1: Overview of the different models presented in this thesis, which can be categorized based on the state space and whether or not the transition probabilities are conditioned on the state of the neighbouring cells.

the data. In Section 4.4, we try to combine both the continuous state space and spatial dependence by conditioning the transition densities on the states of the neighbouring pixels in the CSS SCA model. We search for an aggregation function g that captures the state of the neighbourhood influencing the transition densities.

For all of the models we will show some results of a simulation of a few hours for illustration of the model. A comparison between the models will be made in Chapter 5.

4.1 Reference Model: Markov Chain (MC) with finite number of cloud types

The simplest stochastic parameterization of clouds in Dorrestijn et al. (2013a) is the MC without conditioning on the macroscopic state or neighbouring cells. This will be the reference model for the comparison of all our other models. In Dorrestijn et al. (2013a), the cells are classified into a finite number of cloud types based on cloud top height and column rain fraction. Because the GOES16 satellite product does not give column rain fraction, we classify the cells based on the combination of CTH and COD, using the 10 cloud type classes from the ISSCP classification (Fig. 2.2).

Denote the classified cloud states by $C_i(t) := c(S_i(t))$ which only take discrete values in $\{1, \ldots, N\}$. The transition probability from state $C_i(t) = s$ to $C_i(t + \Delta t) = k$ is

$$p(s, k) = p_{sk} = \mathbb{P}(C_i(t + \Delta t) = k | C_i(t) = s), \tag{4.1}$$

with $\sum_{k=1}^{N} p_{sk} = 1$ for every $s \in \{1, \ldots, N\}$. These transition probabilities determine an $N \times N$ transition matrix.

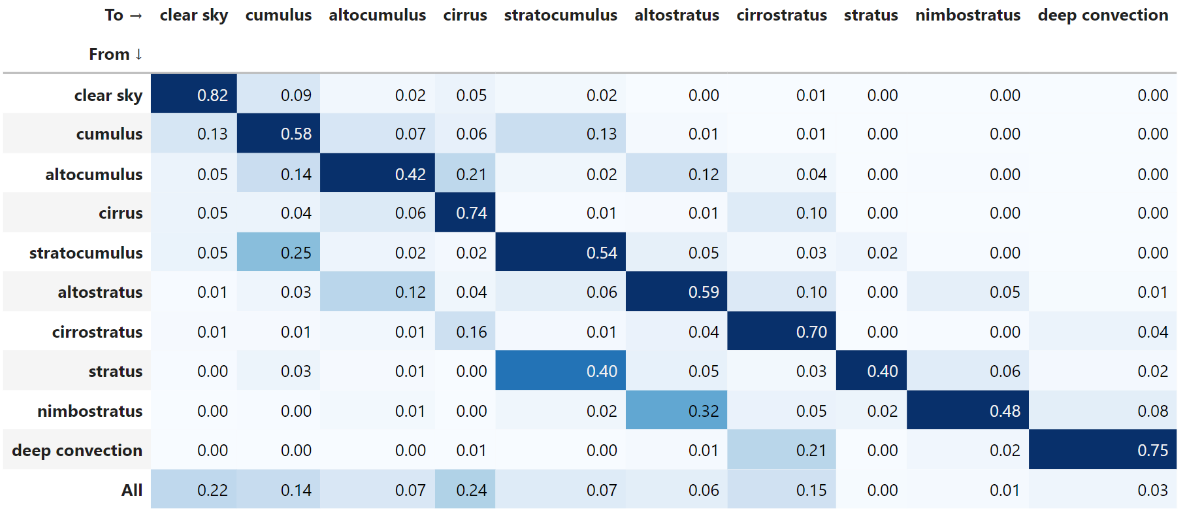
Following the time resolution of the GOES16 satellite data, we take $\Delta t = 10$ minutes. We count transitions in the advection corrected GOES16 satellite data to determine the transition probabilities, using all data at once. This gives the estimate for the transition probability p_{sk} from state s to k by

$$\hat{p}_{sk} = \frac{\sum_{i,t} \mathbb{1}(c_i(t) = s, c_i(t + \Delta t) = k)}{\sum_{i,t} \mathbb{1}(c_i(t) = s)}, \tag{4.2}$$

with $c_i(t)$ the observation of $C_i(t)$ at location i and time t. The estimates are shown in Table 4.1. In the last row we also denote the overall cloud type fractions.

4.1.1 Simulation

As initial state we take an image of our area of interest from the GOES satellite and determine the cloud types at every location i, see Figure 3.5a. For every location we simulate a Markov Chain with transition probabilities from Table 4.1, with time step $\Delta t = 10$ minutes. After only three hours the spatial distribution of the different cloud types looks completely decorrelated (not shown here, but can be seen further on in Figure 5.2a), since of course there was no spatial coupling between any of the pixels. In Figure 4.2 we zoom in into the evolution of the cloud state over time for a fixed location i. The state jumps from one to another state in the discrete state space.



To → From ↓	clear sky	cumulus	altocumulus	cirrus	stratocumulus	altostratus	cirrostratus	stratus	nimbostratus	deep convection
clear sky	0.82	0.09	0.02	0.05	0.02	0.00	0.01	0.00	0.00	0.00
cumulus	0.13	0.58	0.07	0.06	0.13	0.01	0.01	0.00	0.00	0.00
altocumulus	0.05	0.14	0.42	0.21	0.02	0.12	0.04	0.00	0.00	0.00
cirrus	0.05	0.04	0.06	0.74	0.01	0.01	0.10	0.00	0.00	0.00
stratocumulus	0.05	0.25	0.02	0.02	0.54	0.05	0.03	0.02	0.00	0.00
altostratus	0.01	0.03	0.12	0.04	0.06	0.59	0.10	0.00	0.05	0.01
cirrostratus	0.01	0.01	0.01	0.16	0.01	0.04	0.70	0.00	0.00	0.04
stratus	0.00	0.03	0.01	0.00	0.40	0.05	0.03	0.40	0.06	0.02
nimbostratus	0.00	0.00	0.01	0.00	0.02	0.32	0.05	0.02	0.48	0.08
deep convection	0.00	0.00	0.00	0.01	0.00	0.01	0.21	0.00	0.02	0.75
All	0.22	0.14	0.07	0.24	0.07	0.06	0.15	0.00	0.01	0.03

Table 4.1: Transition probabilities between different cloud types. The row **'All'** contains the overall fraction for each cloud type.

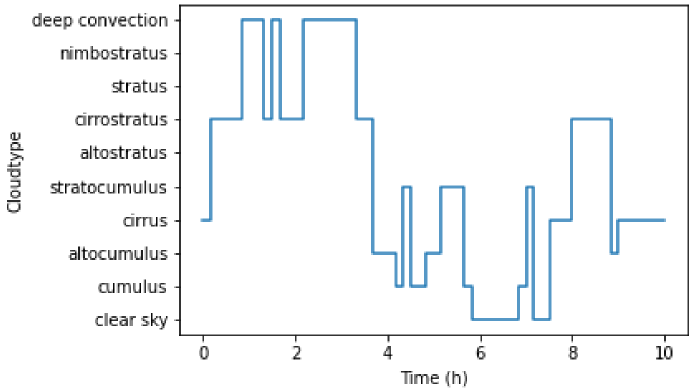


Figure 4.2: One realization of the cloud type evolution at one location over time.

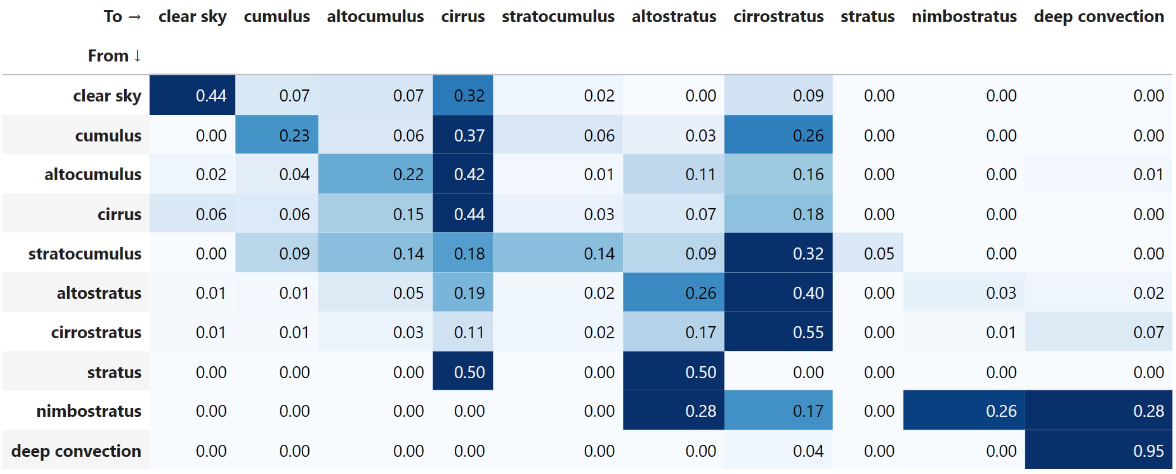
4.2 Reference Model: Stochastic Cellular Automaton (SCA) with finite number of cloud types

In this study we are interested whether we can improve the spatially dependent SCA model from Dorrestijn et al. (2013a). Since we are training the models on a different data set, we also implement the SCA as was suggested in this paper for comparison. Here the transition probabilities $p(s,k)$ are conditioned on a state γ which is a function of the cloud types in the neighbourhood,

$$p_\gamma(s,k) = \mathbb{P}\left(C_i(t+\Delta t) = k \mid C_i(t) = s, g(C_{\{i\}}(t)) = \gamma\right), \tag{4.3}$$

where the neighbourhood $\{i\}$ of a location are the 8 cells surrounding the location. In Dorrestijn et al. (2013a) there are only 5 cloud types; clear sky (CL), shallow cumulus (SC), congestus (CO), deep (DE) and stratiform (ST). They use the following neighbourhood function

$$g\left(C_{\{i\}}(t)\right) = 1 \times \#\{\mathrm{SH}\}_i + 2 \times \#\{\mathrm{CO}\}_i + 3 \times \#\{\mathrm{DE}\}_i + 4 \times \#\{\mathrm{ST}\}_i, \tag{4.4}$$



To →	clear sky	cumulus	altocumulus	cirrus	stratocumulus	altostratus	cirrostratus	stratus	nimbostratus	deep convection
From ↓										
clear sky	0.44	0.07	0.07	0.32	0.02	0.00	0.09	0.00	0.00	0.00
cumulus	0.00	0.23	0.06	0.37	0.06	0.03	0.26	0.00	0.00	0.00
altocumulus	0.02	0.04	0.22	0.42	0.01	0.11	0.16	0.00	0.00	0.01
cirrus	0.06	0.06	0.15	0.44	0.03	0.07	0.18	0.00	0.00	0.00
stratocumulus	0.00	0.09	0.14	0.18	0.14	0.09	0.32	0.05	0.00	0.00
altostratus	0.01	0.01	0.05	0.19	0.02	0.26	0.40	0.00	0.03	0.02
cirrostratus	0.01	0.01	0.03	0.11	0.02	0.17	0.55	0.00	0.01	0.07
stratus	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.00
nimbostratus	0.00	0.00	0.00	0.00	0.00	0.28	0.17	0.00	0.26	0.28
deep convection	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.95

Table 4.2: Transition probabilities $p_\gamma(s,k)$ from cloud type s to k with neighbourhood state $\gamma = 40$.

where $\#\{\cdot|\}_i$ denotes the number of occurences of the cloud type in the neighbourhood of i. Naturally, it follows that always $\#\{\mathrm{CL}\}_i{+}\#\{\mathrm{SH}\}_i{+}\#\{\mathrm{CO}\}_i{+}\#\{\mathrm{DE}\}_i{+}\#\{\mathrm{ST}\}_i = 8$. The function is intuitively constructed trying to assign more weight to a more convectively active neighbourhood. Since in our setup, we have a different classification for the cloud types, we have to adjust the function slightly. We do keep the same structure giving weight to each of the cloud types occurring in the neighbourhood. We define

$$\begin{aligned} g(C_{\{i\}}(t)) = 1\times\#\{\mathrm{Cu}\} + 1\times\#\{\mathrm{Ac}\} + 2\times\#\{\mathrm{Sc}\} + 2\times\#\{\mathrm{As}\} + 3\times\#\{\mathrm{St}\} \quad (4.5) \\ + 4\times\#\{\mathrm{Ns}\} + 5\times\#\{\mathrm{Cb}\} + 6\times\#\{\mathrm{Cs}\} + 7\times\#\{\mathrm{Ci}\}, \quad (4.6) \end{aligned}$$

where the abbreviations used for the cloud types are in Figure 2.2. Dorrestijn et al. (2013a) give these kind of weights to indicate the degree to which the environment is convectively active. The more neighbouring sites are in a state with convection the higher the result of g. The high weights to cirrostratus and cirrus are debatable, but in this case we follow the weights from Dorrestijn. Since there the stratiform cloud type also got the highest weight. All possible values of γ are then within $\{0, 1, 2, \ldots, 56\}$ leading to 57 different transition matrices. The transition probability $p_\gamma(s,k)$ is for all possible states s, k and values of γ are estimated, similar to Eq. (4.2), by

$$\hat{p}_\gamma(s,k) = \frac{\sum_{i,t} \mathbb{1}\left(c_i(t) = s, g(c_{\{i\}}(t)) = \gamma, c_i(t+\Delta t) = k\right)}{\sum_{i,t} \mathbb{1}\left(c_i(t) = s, g(c_{\{i\}}(t)) = \gamma\right)}, \quad (4.7)$$

with $c_{\{i\}}(t)$ the observation of the cloud states in the neighbourhood of i at time t. For example the transition probabilities for $\gamma = 30$ are given in Table 4.2.

4.2.1 Simulation

Starting from the scene in Figure 3.5a, we run the SCA, which results in the realisation shown in Figure 4.3. In this image we zoomed in in a smaller part of of the scene of 300 × 300 km. Every pixel changes according to the transition probabilities corresponding to the specific neighbourhood state. The realisation after only 3 time steps ($3 \cdot 10$ min =

30 min) looks already much more noisy than the starting image. We will explore these effects more thoroughly in the results (Chapter 5 and discussion (Section 6.1).

4.3 Continuous State Space Markov Chain (CSS MC)

The Markov Chain model on a discrete state space only tracks the current cloud type at a location, but we expect that a change from one to another cloud type is much more likely if the CTH or COD is close to a boundary between two cloud types than otherwise. If we have a stratus cloud with very high COD, it is much less likely to break open to stratocumulus, than if the COD is low and close to the boundary with stratocumulus. Therefore we are interested whether it is possible to get a more realistic model when we model the MC in the continuous COD/CTH state space. Also the stochastic model is eventually meant as a parameterization in macroscopic climate models. If we have a realization of the exact state of the cloud, in COD/CTH space, this could give a more direct feedback for the macroscopic model, than evaluating only the cloud type at a location.

For this model we continue with the general model formulation from Chapter 2. We still neglect the dependence on neighbouring cells as was suggested in Eq. (2.4), so the transition density from Eq. (2.6) loses the dependence on the neighbourhood state γ and becomes

$$p(s, k) = p_s \mathbb{1}_{\{(0,-1)\}}(k) + (1 - p_s) f^{(c)}(k|s) \mathbb{1}_{\mathbb{R}\times\mathbb{R}^+}(k). \tag{4.8}$$

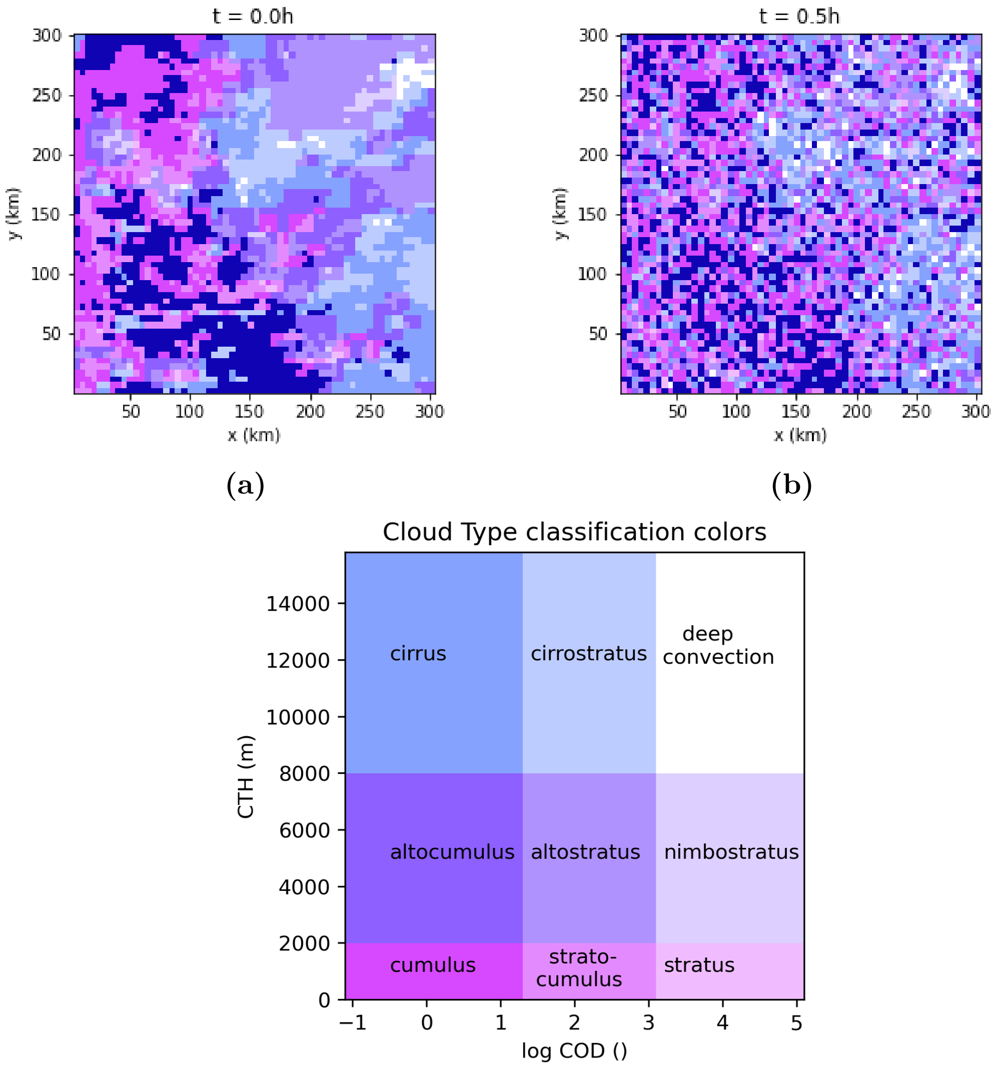
The probability of transitioning to a clear sky state p_s and the cloud transition density $f^{(c)}(k|s)$ depend only on the current state at that location $S_i(t) = s$. Therefore we do not have to search for the specific form of the neighborhood aggregating function g for this model. Now we only need to find a suitable distribution function for the cloud distribution and estimate its parameters and p_s for any state s.

4.3.1 Distribution selection and parameter estimation

The transition density $p(s, k)$ is in fact a conditional distribution on starting state s. Therefore we filter the data for transitions from similar states s and look at the histograms of the states k to which they transitioned. The precise process is described below.

Transition to clear sky

First we consider the estimation of p_s, the probability of transitioning to a clear sky. This is easiest in the case we start from a clear sky, i.e. $s = (0, -1)$. Then we just count the number of transitions to clear sky, which gives just as in the previous model $p_{(0,-1)} = 0.82$, see Table 4.1. If we start from a state which is a cloud, i.e. $s \in \Omega_c$, the estimation of p_s is less straightforward, since the probability of multiple observations with the exact same value for s equals zero. We assume that the behaviour for the state in the next time step is similar for starting states 'close' to one another. The probability of clear sky p_s can be estimated by using a Nadaraya–Watson (NW) estimator (Wand



(a) **(b)**

(c)

Figure 4.3: Snap shots of the simulation of the SCA starting from (a) an original image and (b) after 30 minutes. (c) The color coding for the cloud types, which is the same as before.

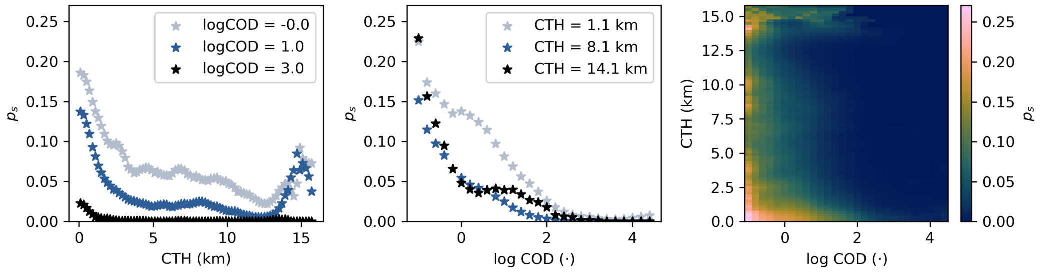


Figure 4.4: Local estimates of p_s using kernel bandwidths $\Delta h = 200$m and $\Delta d = 0.2$. In the left figure the estimates are plotted against the CTH for fixed COD and in the middle figure vice versa. In the right figure all estimates are shown in the CTH-COD space.

and Jones, 1994). Define the random variable $Z_i(t)$ such that $Z_i(t) = 1$ if $S_i(t) = (0, -1)$ and $Z_i(t) = 0$ if $S_i(t) \in \Omega_c$, i.e. $Z_i(t)$ indicates whether the state at a pixel is a clear sky or not. Then $p_s = \mathbb{E}\left[Z_i(t + \Delta t)|S_i(t) = s\right]$. Then the NW estimator for p_s is

$$\hat{p}_s = \frac{\sum_{i,t} K_{\Delta s}(s - s_i(t))z_i(t + \Delta t)}{\sum_{i,t} K_{\Delta s}(s - s_i(t))}, \tag{4.9}$$

with $s_i(t)$ and $z_i(t)$ the observations of $S_i(t)$ and $Z_i(t)$ at location i and time t, and $K_{\Delta s}$ a kernel with bandwidth $\Delta s = (\Delta d, \Delta h)$ giving weight to an observation, with $\Delta d, \Delta h > 0$. Here we will use a boxcar Kernel,

$$K_{(\Delta d, \Delta h)}((h, d)) = \begin{cases} 1 & \text{if } |d| < \Delta d, |h| < \Delta h, \\ 0 & \text{o.w.} \end{cases}, \tag{4.10}$$

giving equal positive weight to observations $s_i(t) = (d_i(t), h_i(t))$ close to $s = (d, h)$.

We calculate these estimates for a finite number of states $s = (d, h)$ with $d = -1, -0.8, -0.6, \ldots, 4.4$ and $h = 100, 300, 500, \ldots, 15700$ and $\Delta h = 200$ and $\Delta d = .2$, and use a nearest neighbour estimation for p_s, with s not belonging to the grid. Since the spacing between calculations of different states is smaller than two times the bandwidths of the kernel, estimations for subsequent states use partially the same observations, smoothing the estimation. The estimates are shown in Figure 4.4. Here we see that for a fixed height the probability for transitioning to clear sky increases for clouds with a low optical depth. This seems reasonable since a state of high optical depth corresponds with more condensed water in the column. It would cost more energy for all the water droplets to evaporate. The transition probability to clear sky also increases for the lower clouds as can be seen in the left panel of the figure, and for low COD also for the most high clouds. The low clouds are cumulus clouds, which are often small (O(100m)), with clear sky in between. It is likely that the next observation, although corrected for advection, is not at the cloud anymore. Also these clouds are likely to resolve, so it can be a mix of noise and signal. For the higher clouds, these are the anvils which are likely to dissolve at some point.

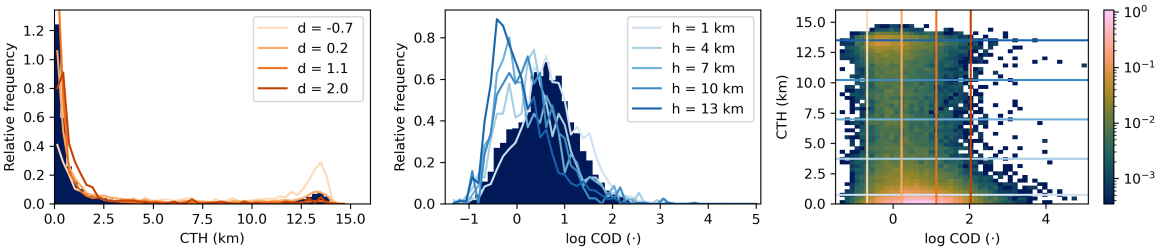


Figure 4.5: Histograms of states $S_i(t+\Delta t)$ conditioned on $S_i(t) = (0, -1)$ is a clear sky and $S_i(t+\Delta t) \in \Omega_c$ is a cloud. The histograms of the marginals (a) $H_i(t+\Delta t)$ and (b) $D_i(t+\Delta t)$ are shown in filled dark blue. In figure (c) the joint histogram is shown. All histograms are normalized such that they integrate to 1. The lines in (a) and (b) show the conditional histograms of the CTH and COD respectively, they are conditioned on the values in the legends.

Transition to cloud

Next we have to determine a model for the cloud transition density $f^{(c)}(k|s)$. To keep our model as simple as possible, we choose one parametric distribution $f^{(c)}(k|\theta(s))$, where the parameter θ will vary based on the starting state s. First we look at the case where $s = (0, -1)$, the histogram of the next states k conditioned that they are a cloud are shown in figure 4.5.

First we ask ourselves whether it is possible to assume that $H_i(t+\Delta t)$ and $D_i(t+\Delta t)$ are independent conditioned on $S_i(t)$. Therefore we look at the conditional densities in Figure 4.5. The histograms of CTH observations $h_i(t)$ given that the corresponding COD observations $d_i(t)$ are close to a fixed COD d, and that the previous state $s_i(t-\Delta t) = (0. - 1)$ was a clear sky pixel are shown as lines, for a couple of values d in Figure 4.5a and vice versa for $d_i(t)$ in Figure 4.5b. If the CTH and COD are indeed independent for the next time step these distributions for different fixed values should be similar. For the CTH in the left panel the lines seem close to one another, except for the peak at higher clouds for low COD. For the COD in the middle panel the major part of the distribution seems to shift to lower values for higher CTH. So a negative correlation is observed, higher clouds tend to have lower COD. Although strictly speaking the assumption that these variables are independent is not valid, we do think it is still useful. This is to keep estimations more simple, without imposing something completely untrue. In later versions of the models this assumption should be reconsidered though. But for now we assume

$$f^{(c)}((d,h)|s) = f_{D|S}(d|s) \cdot f_{H|S}(h|s), \tag{4.11}$$

with $s = (0, -1)$, where for simplicity of notation $D|S$ denotes the COD state in the next time step given the state in the previous step, i.e. $D_i(t+\Delta t)|S_i(t)$, and analogously for $H|S$ the CTH state.

Based on the shape of the histogram in Figure 4.5b, it is assumed that the log of cloud optical depth is normally distributed, i.e.

$$D|S = (0, -1) \sim \mathcal{N}(\mu_{(0,-1)}, \sigma^2_{(0,-1)}), \tag{4.12}$$

with mean $\mu_{(0,-1)}$ and variance $\sigma^2_{(0,-1)}$. These parameters are estimated using the most simple composite maximum likelihood Varin et al. (2011), formally maximizing the likelihood constructed as if all observations were independent. This likelihood is also referred to as the pseudolikelihood. We try to mimic this independence by using only observations which are spatially distanced.

The estimators of the mean and variance become respectively

$$\hat{\mu}_{(0,-1)} = \frac{1}{n} \sum_{\substack{i,t \\ S_i(t)=(0,-1) \\ S_i(t+\Delta t)\in\Omega_c}} D_i(t+\Delta t), \tag{4.13}$$

$$\hat{\sigma}^2_{(0,-1)} = \frac{1}{n} \sum_{\substack{i,t \\ S_i(t)=(0,-1) \\ S_i(t+\Delta t)\in\Omega_c}} \left(D_i(t+\Delta t) - \hat{\mu}_{(0,-1)}\right)^2, \tag{4.14}$$

where n is the number of transition observations (i,t) starting from clear sky and going to a cloud.

The choice for a suiting probability distribution of the next CTH state $H|S$ is more difficult, since the values of CTH are bounded by the ground and the Tropopause (at 15-17 km height). The distribution is not bell-shaped, but really cut-off at the lower boundary, so a normal distribution would not suit. We suggest a Beta distribution (see Appendix A.1), which has in its standard form a bounded domain on [0, 1]. But we can of course map the CTH to this domain, using $H' = H/h_{max}$, with h_{max} the maximum CTH. The Beta distribution is usually uni modal, unless both shape parameters α and β are smaller than 1, in which case the density is unbounded at the boundaries. We do however not observe one peak for the next state CTH but as can be seen in the histogram from Figure 4.5a, there are is a bimodality. We observe one peak close to the surface and on for high clouds with CTH at ∼1 km height. Therefore we also try to fit a mixture of two Beta distributions, i.e.

$$H'|S = (0,-1) \sim q_{(0,-1)}\text{Beta}(\mu_{1,(0,-1)}, \nu_{1,(0,-1)}) + (1 - q_{(0,-1)})\text{Beta}(\mu_{2,(0,-1)}, \nu_{2,(0,-1)}), \tag{4.15}$$

where $1 \geq q_{(0,-1)} \geq \frac{1}{2}$ is the weight on the first Beta distribution, and the mean μ and shape parameter ν are a different parameterization for the Beta-distribution than is used normally, which is discussed in Appendix A.1. $q_{(0,-1)}$ is chosen to be always larger than $\frac{1}{2}$, such that the first Beta distribution with parameters μ_1 and ν_1 is the dominant mixture component.

Again the parameters, in this case $\phi_{(0,-1)} = \left(q_{(0,-1)}, \mu_{1,(0,-1)}, \nu_{1,(0,-1)}, \mu_{2,(0,-1)}, \nu_{2,(0,-1)}\right)$, are estimated by the maximizing the pseudolikelihood

$$\mathcal{L}(\phi_{(0,-1)}; (\mathbf{d}, \mathbf{h})) = \prod_{i,t} f_{H|S}\left(h_i(t+\Delta t)|\phi_{(0,-1)}\right)^{\mathbb{1}(s_i(t)=(0,-1), s_i(t+\Delta t)\neq(0,-1))}, \tag{4.16}$$

where $(\mathbf{d}, \mathbf{h})$ denote the state observations for all i and t of the COD and CTH. In contrast to the normal distribution for the COD, there is no closed-form expression for these parameter estimates. In those cases often the Method of Moment (MoM) Estimator, which in many cases does still have an explicit form is used as an initial guess for the ML estimator. After which an algorithm tries to improve the estimation, with respect to the likelihood function. However, for the Beta mixture there is no explicit MoM estimator. We use the Nelder-Mead optimization method for maximizing the log-likelihood function. The initial guesses for the parameters are crucial for convergence of the algorithm to the global optimum, therefore we have to choose them wisely. As initial guesses we take,

$$\begin{aligned}
\hat{q}_{(0,-1)} &= 0.8, && (4.17)\\
\hat{\mu}_{1,(0,-1)} &= \bar{h}/h_{max}, && (4.18)\\
\hat{\nu}_{1,(0,-1)} &= 20, && (4.19)\\
\mu_{2,(\hat{0},-1)} &= 1 - \hat{\mu}_{1,(0,-1)}, && (4.20)\\
\nu_{2,(\hat{0},-1)} &= 20, && (4.21)
\end{aligned}$$

where $\bar{h}$ denotes the mean CTH of the observations from clear sky to a cloud state. Taking the sample mean as the mean for the main mode, gives a good initial guess for the location of this dominant mode. Since there is often a bimodality, we set the mean of the second component to the other side of the domain as initial guess.

The last choice in our model is $f^{(c)}(k|s)$ with $s \in \Omega_c$, i.e. s is a cloud. We will check whether it is reasonable to assume the same distribution as for $s = (0, -1)$ only with different estimates for the parameters. For a finite number of states s in Ω_c we check the histograms of observations $s_i(t + \Delta t)$, with i, t such that the previous state is close to s and the state itself is a cloud,

$$K_{\Delta s}(s - s_i(t)) = 1 \quad \text{and} \quad s_i(t + \Delta t) \in \Omega_c, \tag{4.22}$$

where K is the kernel from Eq. (4.10). Similar to Figure 4.5, a few are shown in Figure 4.6 and more in Appendix B.1. All the 2d histograms seem to indicate some kind of ellipse shaped distribution, with the main axis in the CTH or COD coordinate, therefore we assume again that the random variables $D_i(t + \Delta t)$ and $H_i(t + \Delta t)$ are independent conditioned on $S_i(t)$. Also the distribution of the cloud optical depth looks normal and the CTH still has a bimodality, although the second mode is not prominent. But the fit for a Beta mixture is again much better than a single Beta distribution for these few examples. These observations are only made for a few choices for the starting state s within the cloud state space. We will check whether the choice for a Beta mixture is favourable above the single Beta, and whether the ML estimate converges indeed to a distribution which represents the data better. But instead of looking at the histograms for many options of s, we will use a measure to determine the goodness of fit for the estimated distributions.

The maximimum composite likelihood estimator for ϕ_s maximizes the composite likelihood,

$$\mathcal{L}(\phi_s; (\mathbf{d}, \mathbf{h})) = \prod_{i,t} f_{H|S}\left(h_i(t + \Delta t)|\phi_s\right)^{w_i \mathbb{1}(s_i(t+\Delta t)\neq(0,-1))}, \tag{4.23}$$

with non-negative weights $w_i = K_{\Delta s}(s - s_i(t))$, according to the used kernel, which again will be the Box car kernel. If the observation $s_i(t)$ is close to s the weight is higher.

There are multiple measures to determine how well a distribution models a data sample. Here we will show the results of the Akaike information criterion (AIC), which is a measure to select the a model from multiple models and is defined as

$$AIC = 2k - 2\log(\hat{L}), \tag{4.24}$$

where k is the number of parameters in the model and $\hat{L}$ the likelihood of the model with estimated parameters. The criterion penalizes for more parameters and rewards a high likelihood, the model with the lowest AIC value is preferred over the other models. In Figure 4.7 the AIC values for the Beta mixture model and single Beta model are shown. It can be observed that although the Beta mixture contains more parameters the AIC scores are lower or equal than for the single Beta for the estimated distributions at all values for s. In the figure only the fits from cloud to cloud are shown, but the same difference is observed for transitions from the clear sky state to a cloud. The related and frequently used as well Bayesian information criterion (BIC), also gives preference to the Beta mixture.

Therefore we come to the same parametric distribution for the next step cloud state from a clear sky as from a cloud resulting, namely

$$\begin{aligned} f^{(c)}(k|s) &= f^{(c)}((d,h)|\theta(s)) && (4.25)\\ &= f_{D|S}(d|\theta(s)) \cdot f_{H|S}(h|\theta(s)) && (4.26)\\ &= f_{D|S}(d|\theta(s)) \cdot f_{H'|S}\left(h/h_{max}|\theta(s)\right), && (4.27) \end{aligned}$$

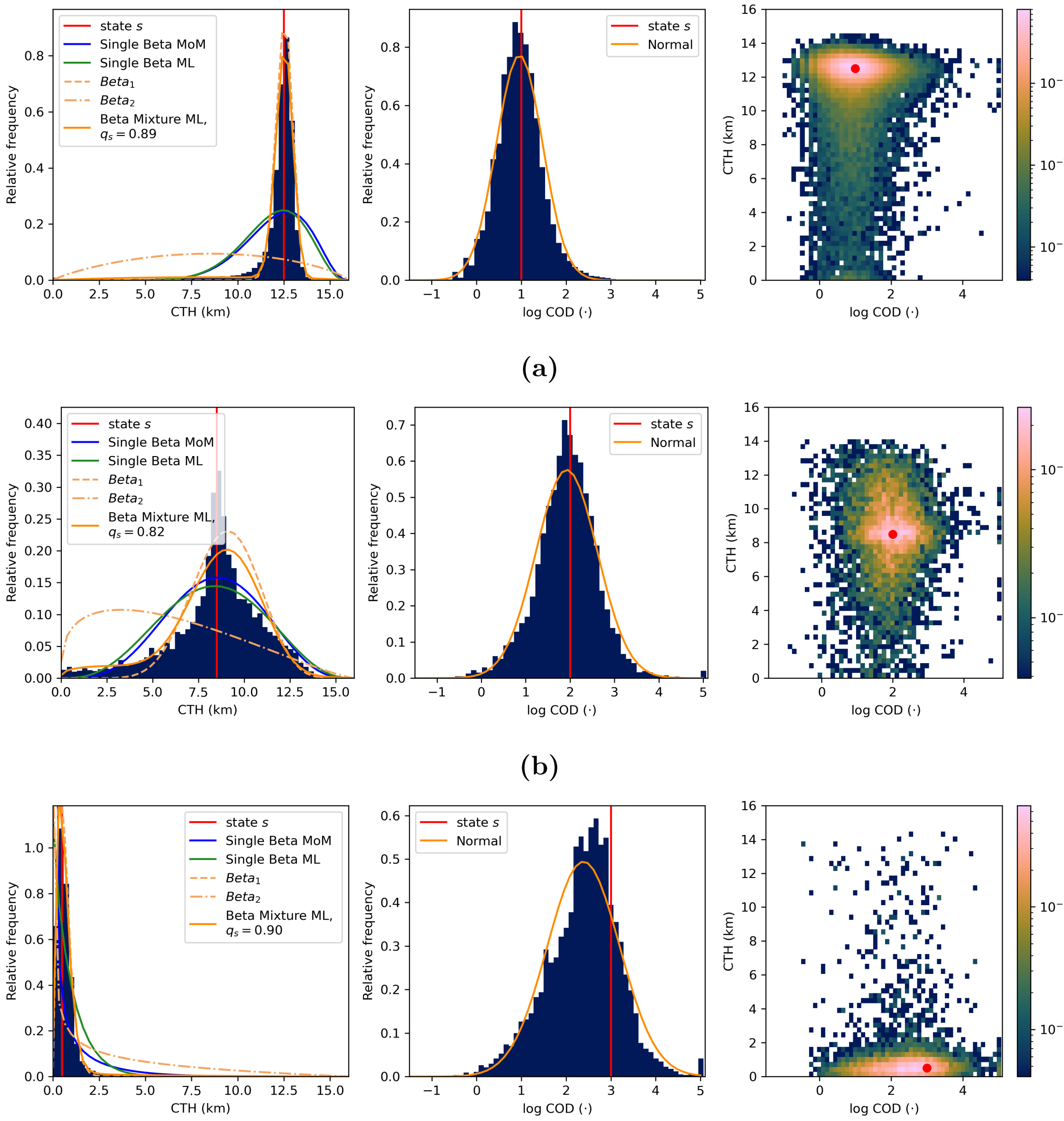
with

$$\begin{aligned} \theta(s) &= (\mu(s), \sigma(s), q(s), \mu_1(s), \nu_1(s), \mu_2(s), \nu_2(s)), && (4.28)\\ D|(S=s) &\sim \mathcal{N}(\mu(s), \sigma(s)), && (4.29)\\ H'|(S=s) &\sim q(s)\text{Beta}(\mu_1(s), \nu_1(s)) + (1-q(s))\text{Beta}(\mu_2(s), \nu_2(s)), && (4.30) \end{aligned}$$

where all the parameters are a function of the starting state s, which are relatively smooth due the use of the Box Car kernel. In Figure 4.8, we see the estimated values for a few of the parameters. The estimates for all variables are shown in Appendix B.2.

4.3.2 Simulation

Using the transition density derived in Section 4.3.1 we perform again a simulation starting from the same original image as with the previous model, see Figure 3.5a. In this case the state of a pixel contains both a CTH and COD value instead of only the cloud type in the previous model. Since there was no spatial coupling included in this model yet, no spatial structure will be simulated by this model. A realization of the state $S_i(t)$ for one location i at $t = 0, .., 10$h with time steps of Δ =10 min is shown in Figure 4.9. Also the cloud type, which is calculated from the (COD, CTH) state is shown again.

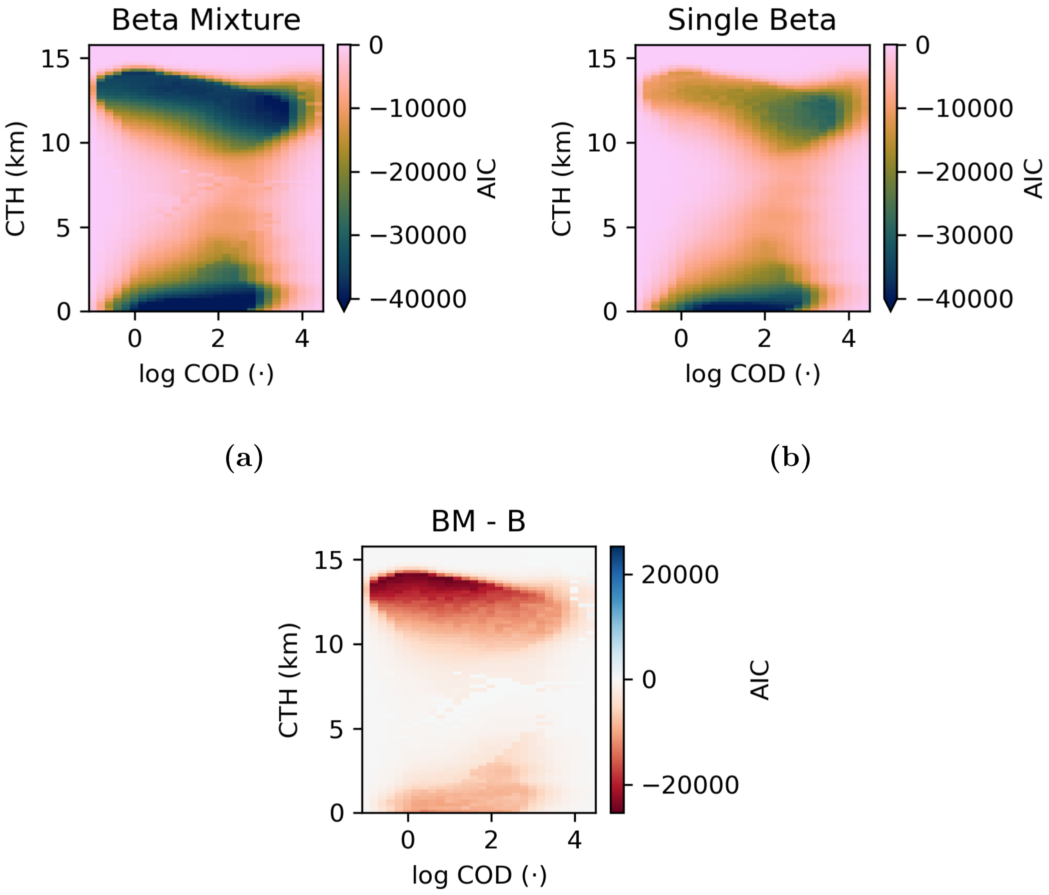


(a)

(b)

(c)

Figure 4.6: Histograms of states $s_i(t + \Delta t) \in \Omega_c$, with i, t such that the previous state $s_i(t)$ is close to s, i.e. $K_{(\Delta d, \Delta h)}(s - s_i(t)) = 1$ with (a) $s = (1, 12000)$, (b) $s = (2, 9000)$ and (c) $s = (3, 1000)$, with $\Delta h = 200$m and $\Delta d = 0.2$. The value of s is indicated by the red lines and dot. The fitted distributions as described in text are also shown.

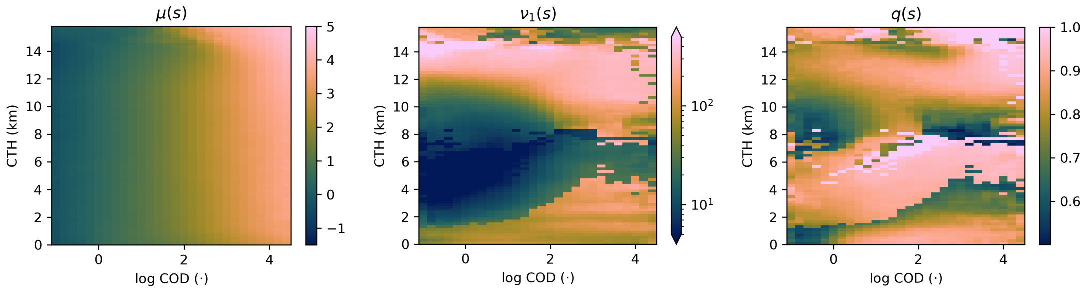


(a)

(b)

(c)

Figure 4.7: The AIC scores for the ML estimators for (a) Beta mixture and (b) single Beta distribution for the next state CTH from a cloud state $s = (d, h)$. (c) The difference in the AIC scores for the two distributions.



(a)

(b)

(c)

Figure 4.8: Estimates $\mu(s), \nu_1(s)$ and $q(s)$ from Eq. (4.29) and (4.30) with bin widths $\Delta h = 200$m and $\Delta d = 0.2$.

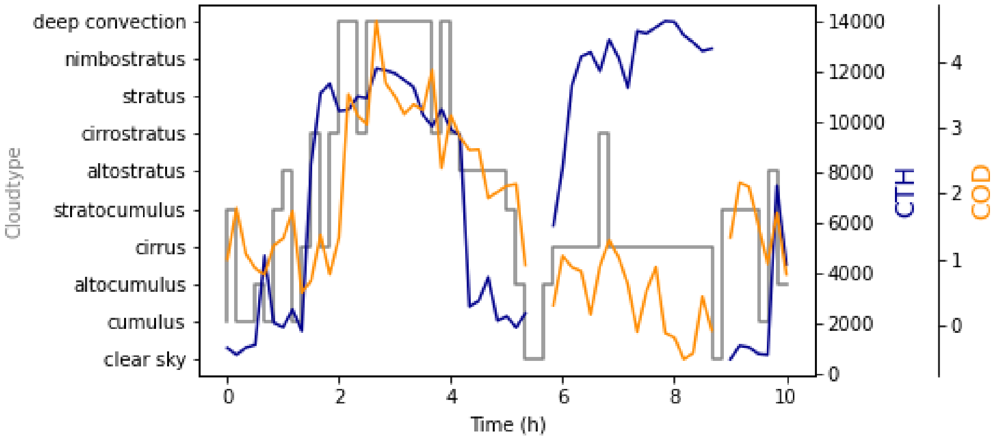


Figure 4.9: Realization of the cloud state $S_i(t)$ for one location i. The state consists of a CTH (blue) and COD (yellow), corresponding to a cloud type. The CTH/COD are not shown in case there was a clear sky at the location.

4.4 Continuous State Space Stochastic Cellular Automaton (CSS SCA)

In this section we will combine the stochastic MC with the conditioning on the cloud state in neighbouring cells. For the function determining the state of the neighbourhood Dorrestijn et al. (2013a) did not research multiple options. They constructed a function based on physical intuition. In the following sections we will try to find the best option from a selection of possible features, describing the neighbourhood state.

4.4.1 Selection of additional spatial variable

For this first model which includes conditioning on the neighbourhood we define the neighbourhood $\{i\}$ to be the 8 grid points around the location i. Taking the state at each of these eight grid points as an extra variable to condition the transition densities on, would lead to many different possible states, for all of which we would need enough data. Moreover, it is expected that many of these states, would lead to the same information regarding the distribution of the next time step cloud state. For example the orientation is not expected to have much influence. To reduce the high dimensionality of the model and make it easier to interpret and understand we will perform feature selection on the neighbourhood states. There are multiple methods possible to select variables deduced from the states in the neighbourhood which influence the transition density $p_\gamma(s, k)$ most.

Feature selection methods can be divided into three different categories (Doquire and Verleysen, 2013). The first category contains wrappers. These methods select the best set of features based on the performance of the model. These methods are therefore expected to result in a good set of features, but can be computationally expensive since multiple models have to be made. Secondly, filters can be used for feature selection. These methods can be performed before the algorithm or model training, during the data preprocessing. Among these filters belong the frequently used correlation coefficients (Hall, 2000), and mutual information (Battiti, 1994). The last category of selection methods consist of algorithms which simultaneously optimize the model and select features, e.g. Lasso regularization (Tibshirani, 1996).

Since our models are not optimizing some objective to fit the model to, we will use a filter as feature selection method. We will use *mutual information* as measure for the selection of features which give the most extra insight about the distribution of the next time step, next to the state of the cloud in the current time step. Mutual information has the advantage over the use of a correlation coefficient that non-linear relationships can be found using this metric. Intuitively said, mutual information gives a measure for the dependency between two variables. When two random variables X and Y are independent, the mutual information between them equals zero. The more 'information' Y contains about X and vice versa, the higher the mutual information between the two variables is. We will use this to determine which spatial variables give the most information about the state of a pixel in the next time step.

Mutual Information

The exact definition for mutual information is as follows, and dates back to Shannon (1948). Given two continuous random variables X and Y with joint density $f_{X,Y}(x,y)$, the *mutual information* $I(X,Y)$ between the two variables is defined as

$$I(X,Y) = \int\int f_{X,Y}(x,y)\log\left(\frac{f_{X,Y}(x,y)}{f_X(x)f_Y(y)}\right)dxdy, \tag{4.31}$$

where $f_X(x)$ and $f_Y(y)$ denote the marginal densities of X and Y respectively. Note that if X and Y are independent $f_{X,Y}(x,y) = f_X(x)f_Y(y)$ and therefore

$$\log\left(\frac{f_{X,Y}(x,y)}{f_X(x)f_Y(y)}\right) = \log(1) = 0. \tag{4.32}$$

So in that case the mutual information equals indeed 0. Futhermore, from Jensen's Inequality it follows that $I(X,Y) \geq 0$.

Estimating the mutual information from our observations is not very straightforward. We do not know the true marginal and joint distributions of any of the variables we will observe, so these have to be estimated. Typically, discretization with equidistant binning is the most straightforward method, and although different methods with e.g. adaptive bin widths, different kernels, or parametric estimations are also developed (o.a. Darbellay, 1999; Tsimpiris et al., 2012; Walters-Williams and Li, 2009), we prefer to go for equidistant binning. This is easy to implement and for a large number of observations still applicable (as in e.g. Brown et al., 2012). The higher computational costs in case of many different variables is not a problem since we need to calculate all the mutual informations only once. This straightforward method is already quite old, so following Moddemeijer (1989) the estimation is made by approximating the densities by histograms. The state space of X and Y are discretized into cells with centers (x_i, y_j). The probability that we observe (x,y) in a bin (x_i, y_j) is denoted by p_{ij}. This probability is then estimated by the number of observations in this bin, divided by the total number of observations. The mutual information is then approximated by

$$I(X,Y) \approx \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{p_i p_j}\right), \tag{4.33}$$

where $p_i = \sum_j p_{ij}$ and $p_j = \sum_i p_{ij}$ are the estimates for the marginal distributions.

Candidate spatial variables

The first candidate spatial variable which comes to mind is the clear sky fraction of the neighbourhood. We expect that the number of adjacent cells which do not contain a cloud is relevant for the state of a pixel. If it is surrounded by only cloud pixels, the probability on clear sky should be really small for example. The clear sky fraction $Z_{\{i\}}(t)$ in the neighbourhood of location i at time t is defined by

$$Z_{\{i\}}(t) = \frac{1}{\#\{i\}} \sum_{j\in\{i\}} Z_j(t), \tag{4.34}$$

where $\#\{i\}$ denotes the number of sites in the neighbourhood of i (8 in our case). Recall that $Z_i(t)$ was defined in Section 4.3.1 to indicate whether there is a clear sky at the pixel. Two other variables which we expect to influence the evolution of a cloud are the mean COD and mean CTH, or actually the difference between these means and the current COD/CTH at a location. We define the centered mean neighbourhood COD and -CTH

$$D_{\{i\}}(t) = \frac{\sum_{j\in\{i\}} D_j(t)\mathbb{1}_{\Omega_c}(S_j(t))}{\sum_{j\in\{i\}} \mathbb{1}_{\Omega_c}(S_j(t))} - D_i(t), \tag{4.35}$$

$$H_{\{i\}}(t) = \frac{\sum_{j\in\{i\}} H_j(t)\mathbb{1}_{\Omega_c}(S_j(t))}{\sum_{j\in\{i\}} \mathbb{1}_{\Omega_c}(S_j(t))} - H_i(t)\mathbb{1}_{\Omega_c}(S_i(t)), \tag{4.36}$$

with $S_i(t) = (D_i(t), H_i(t))$. Notice that we only take the mean over the sites that contain a cloud and we subtract the COD and CTH at the site itself only in the presence of a cloud. If $S_i(t) = (0, -1)$, $D_{\{i\}}(t)$ is simply the mean COD.

Choice

Many different algorithms have been constructed to select a number of relevant feature variables from a set of variables based on mutual information, (Vergara and Estévez, 2014). Here we will calculate the mutual information between the different spatial variables and the next cloud state, to determine which spatial variables $g(S_{\{i\}})(t)$ could improve the model best. We are able to calculate all the different combinations between variables, since we have a limited set of candidate variables. In the models the distribution of $D_i(t + \Delta t)$ and $H_i(t + \Delta t)$ are assumed to be independent conditioned on $S_i(t)$ and $S_{\{i\}}(t)$. Therefore we study the influence of the spatial candidate variables on both these variables separately. Furthermore we setup the models previously such that we estimate the parameter $p_{\gamma,s}$, the probability on a clear sky pixel in the next time step, separately from the cloud transition distribution for the COD and CTH in the next time step. Therefore we also use the mutual information between the spatial variables at time t and $Z_i(t+\Delta t)$. We will use the latter case to explain our method. We calculate the mutual information

$$I\left(Z_i(t+\Delta t), [g(S_{\{i\}}(t)), S_i(t)]\right), \tag{4.37}$$

using the approximation of Eq. (4.33) for different choices of g. The results are shown in Table 4.3 for the possible combinations of mean neighborhood states.

Consider for example $X = Z_i(t+\Delta t)$ and $Y = (S_i(t), Z_{\{i\}}(t))$, the first bold row from Table 4.3. The mutual information $I(X,Y)$ is calculated as follows. First the probability p_{ij} is determined by

$$p_{ij} = \frac{1}{n}\sum_{i',t} \mathbb{1}\left(\left[z_{i'}(t+\Delta t), s_{i'}(t), z_{\{i'\}}(t)\right] \in b_{ij}\right), \tag{4.38}$$

where $z_{i'}(t+\Delta t), s_{i'}(t), z_{\{i'\}}(t)$ are the observations of $Z_{i'}(t+\Delta t), S_{i'}(t), Z_{\{i'\}}(t)$ at time t and location i' respectively, n is the number of observations and b_{ij} are the bins of the range of the variables such that

$$b_{ij} = \{(z', d, h, z) | z' \in b_i, (d, h, z) \in b_j\}, \tag{4.39}$$

where b_i and b_j are the i^{th} and j^{th} bin in the range of X and Y. For example since X takes only the values 0 and 1 in this example $b_0 = \{0\}$ and $b_1 = \{1\}$, resulting in only 2 bins. For the continuous variables we divide the domain into 100 intervals. Using these p_{ij} we can determine $I(X,Y)$ using equation (4.33) and the relations $p_i = \sum_j p_{ij}$ and $p_j = \sum_i p_{ij}$.

Note that these values are the estimated mutual informations based on the discretization of the continuous random variables. For continuous variables the mutual information with the variable itself is not defined, as the joint density w.r.t. the Lebesgue measure does not exist, but for discrete ones it is. By using bins to estimate the mutual information we actually made a discrete version of the continuous variable. The mutual information of a discrete random variable with itself is larger than with any other variable so it gives a maximum to the estimated mutual information. So as a reference the mutual information of the (discretized) $Z_{\{i\}}(t)$ with itself is also shown. It can be seen that the neighbourhood variables together with the current state contain more information about $Z_i(t+\Delta t)$ than the current state $S_i(t)$ on its own. Also note that the clear sky fraction in the neighbourhood contains more information than the mean CTH and COD together. For now we choose only the clear sky fraction in the neighbourhood as extra variable to determine the $p_{\gamma,s}$.

Similarly we look at the mutual information between the features and the next state CTH $H_i(t+\Delta t)$. Here we see that the variables seem to contain less information in general compared to the maximum possible information of 4.1. For our first simple model, we extend the conditioning with only one extra variable, in that case $H_{\{i\}}(t)$ is the best choice. But in a next version taking both the mean CTH and COD could be a good option to obtain significantly more information.

Analogously, for the next time step COD state we choose to take only the mean COD into account for this first model.

X	Candidate features (Y)	Information $I(X, Y)$
$Z_i(t + \Delta t)$	$Z_i(t + \Delta t)$	0.538
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	0.359
	$[S_i(t), H_{\{i\}}(t), Z_{\{i\}}(t)]$	0.351
	$[S_i(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	0.350
	$[\mathbf{S_i(t)}, \mathbf{Z_{\{i\}}(t)}]$	**0.343**
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t)]$	0.306
	$[S_i(t), D_{\{i\}}(t)]$	0.292
	$[S_i(t), H_{\{i\}}(t)]$	0.287
	$S_i(t)$	0.274
$H_i(t + \Delta t)$	$H_i(t + \Delta t)$	4.152
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	1.915
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t)]$	1.796
	$[S_i(t), H_{\{i\}}(t), Z_{\{i\}}(t)]$	1.659
	$[S_i(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	1.564
	$[\mathbf{S_i(t)}, \mathbf{H_{\{i\}}(t)}]$	**1.550**
	$[S_i(t), D_{\{i\}}(t)]$	1.448
	$[S_i(t), Z_{\{i\}}(t)]$	1.343
	$[S_i(t)]$	1.246
$D_i(t + \Delta t)$	$D_i(t + \Delta t)$	4.204
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	1.413
	$[S_i(t), H_{\{i\}}(t), D_{\{i\}}(t)]$	1.273
	$[S_i(t), D_{\{i\}}(t), Z_{\{i\}}(t)]$	1.152
	$[S_i(t), H_{\{i\}}(t), Z_{\{i\}}(t)]$	1.062
	$[\mathbf{S_i(t)}, \mathbf{D_{\{i\}}(t)}]$	**1.029**
	$[S_i(t), H_{\{i\}}(t)]$	0.941
	$[S_i(t), Z_{\{i\}}(t)]$	0.849
	$[S_i(t)]$	0.753

Table 4.3: Mutual informations $I(X, Y)$

4.4.2 Estimating parameters with conditioning on spatial variable

For this model we assume the same distribution for the next step cloud state as in the CSS MC, but now the parameters p_s and $\theta(s)$ also depend on the state of the neighbourhood $\gamma = (\bar{d}, \bar{h}, \bar{z})$, where $\bar{d}$ is the mean COD, $\bar{h}$ the mean CTH and $\bar{z}$ the cloud top height. Then the transition distribution is

$$p_\gamma(s, k) = p_{\gamma,s} \mathbb{1}_{\{(0,-1)\}}(k) + (1 - p_{\gamma,s})\, f^{(c)}(k \mid \theta(\gamma, s)) \mathbb{1}_{\mathbb{R}\times\mathbb{R}^+}(k), \tag{4.40}$$

with $f^{(c)}$ as in (4.25), again a combination of a Beta mixture and normal distribution for the CTH and COD.

The estimation of the parameters $p_{\gamma,s}$ and $\theta(\gamma, s)$ is done similarly as in Eq. (4.9). Only here the kernel should also consider whether the observed neighbourhood state $\gamma_i(t)$ is close to γ. More specifically, since we choose that the transition probability to clear sky only depends on the last element of γ we get

$$\hat{p}_{\gamma,s} = \hat{p}_{\bar{z},s} = \frac{\sum_{i,t} K_{\Delta z,\Delta s}\left(\bar{z} - z_{\{i\}}(t), s - s_i(t)\right) z_i(t + \Delta t)}{\sum_{i,t} K_{\Delta z,\Delta s}\left(\bar{z} - z_{\{i\}}(t), s - s_i(t)\right)}, \tag{4.41}$$

where $z_{\{i\}}(t)$ is the observation of $Z_{\{i\}}(t)$ and $\Delta z > 0$, here we will use again the boxcar kernel,

$$K_{\Delta z,\Delta s}(\bar{z}, (h, d)) = \begin{cases} 1 & \text{if } |d| < \Delta d, |h| < \Delta h, |\bar{z}| < \Delta z, \\ 0 & \text{o.w.} \end{cases}. \tag{4.42}$$

We again estimate the this probability for a grid of values for $\bar{z}$ and s. For states in between the grid, we determine the estimate using nearest neighbours. Similarly we determine the estimated parameters for the transition densities for COD and CTH but then the kernel has an extra condition on the observed central mean COD $\bar{d}$ and CTH $\bar{h}$ respectively instead of the clear sky fraction $\bar{z}$.

4.4.3 Simulation

We run again a simulation starting from the same image (Fig. 3.5a) as in the previous models. In this case, for every time step we take the following steps;

1. we determine neighbourhood states γ for every location i;
2. lookup estimates for $p_{\gamma,s}$ and $\theta(\gamma, s)$, for every location i with state $S_i(t) = s$ and $g\left(S_{\{i\}}\right)(t) = \gamma$. More specifically, say for a location i and time t we have $S_i(t) = s$, and neighbourhood states $Z_{\{i\}}(t) = \bar{z}$, $D_{\{i\}}(t) = \bar{d}$ and $H_{\{i\}}(t) = \bar{h}$. Then $g\left(S_{\{i\}}\right)(t) = \gamma = (\bar{d}, \bar{h}, \bar{z})$. The estimates for $p_{\gamma,s}$ only depend on the clear sky fraction in the neighbourhood $\bar{z}$ and s. Then for the parameters of the next state COD, $\mu(\gamma, s)$ and $\sigma(\gamma, s)$, these are actually only a function of $\bar{d}$ and s and the remaining variables in $\theta(\gamma, s)$ for the next state CTH, are then only a function of $\bar{h}$ and s;
3. draw from the distribution as in equation 4.40 with the corresponding parameters for each location to obtain realizations for $S_i(t + \Delta t)$;

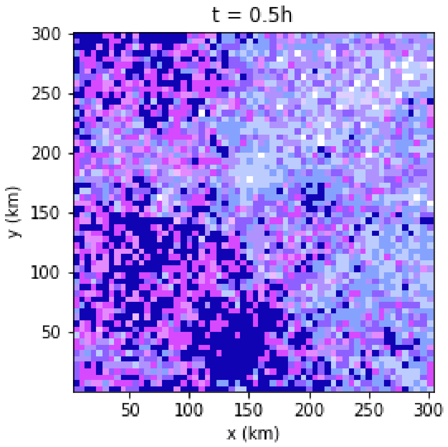


Figure 4.10: One realization of the CSS SCA after 30 minutes, i.e. three time steps of 10 minutes. The color codes for the different cloud types are again the same as in Figure 3.3b.

4. repeat.

In figure 4.10 a realization of one simulation after 30 minutes is shown for illustration. A more detailed comparison of the models simulations will be shown in the Results (Chapter 5).

5

Results and Tuning

In this Chapter we show the simulation results of the models described in the previous Chapter. In particular, we look at the simulated filling fractions and the spatial organization of the cloud types.

5.1 Results

5.1.1 Cloud type fractions & Spatial Organization

In Figure 5.2 realizations of all four models are shown. Each of the simulations start from the same observed image from the real data at $t = 0$. After each time step of $\Delta t = 10$ minutes, new cloud types, in the case for the discrete models, or new states for the COD and CTH for the continuous state space models are simulated. These states are then also mapped to the corresponding cloud types. In the figures the cloud types for all these simulations are shown at four different times. For comparison the observed cloud types following the starting images are shown in Figure 5.1. In Appendix C.1 the realizations are shown on a small part of the simulated domain. In the starting image we observe very large fields of clear sky and cirrus areas. We see that for the MC model, Figure 5.2a, these large structures are dissipating after only 30 minutes already. After 5 hours the distribution of the different cloud types looks very 'random' in the sense that neighbouring pixels look completely uncorrelated. For this model this makes sense, since no spatial dependence is incorporated in this model at all.

For the simulation of the SCA, Figure 5.2b, the evolution of the cloud types looks quite similar to the cloud types simulated by the MC. Again the large structures vanish after a few hours and the image at $t = 5$h has a granular structure. But in this case it seems that after half an hour the structures from the beginning are still more present than in case of the MC. For example, we see larger areas of clear sky remaining. Also the granular state at which we arrive at $t = 5$h seems to have a different composition of cloud types than for the Markov Chain. To see this difference better the cloud type fractions after some time will be explored at the end of this section.

In Figure 5.2c the simulation for the CSS MC is shown. Here we observe again the development of a granular image after only a couple of hours. It seems like the original

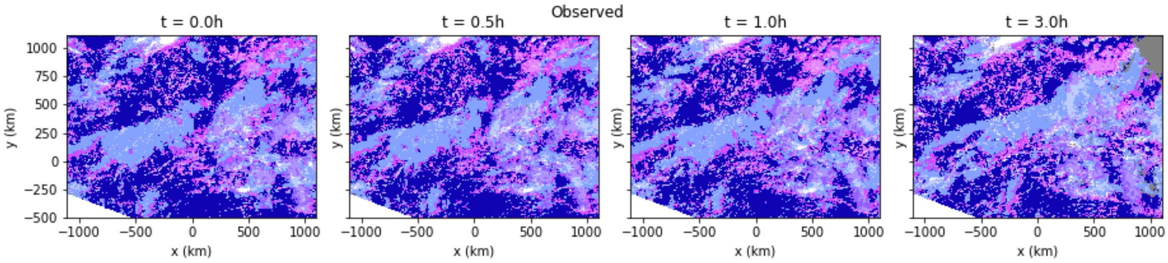


Figure 5.1: Observed development of cloud types from the starting images that was used for the simulations.

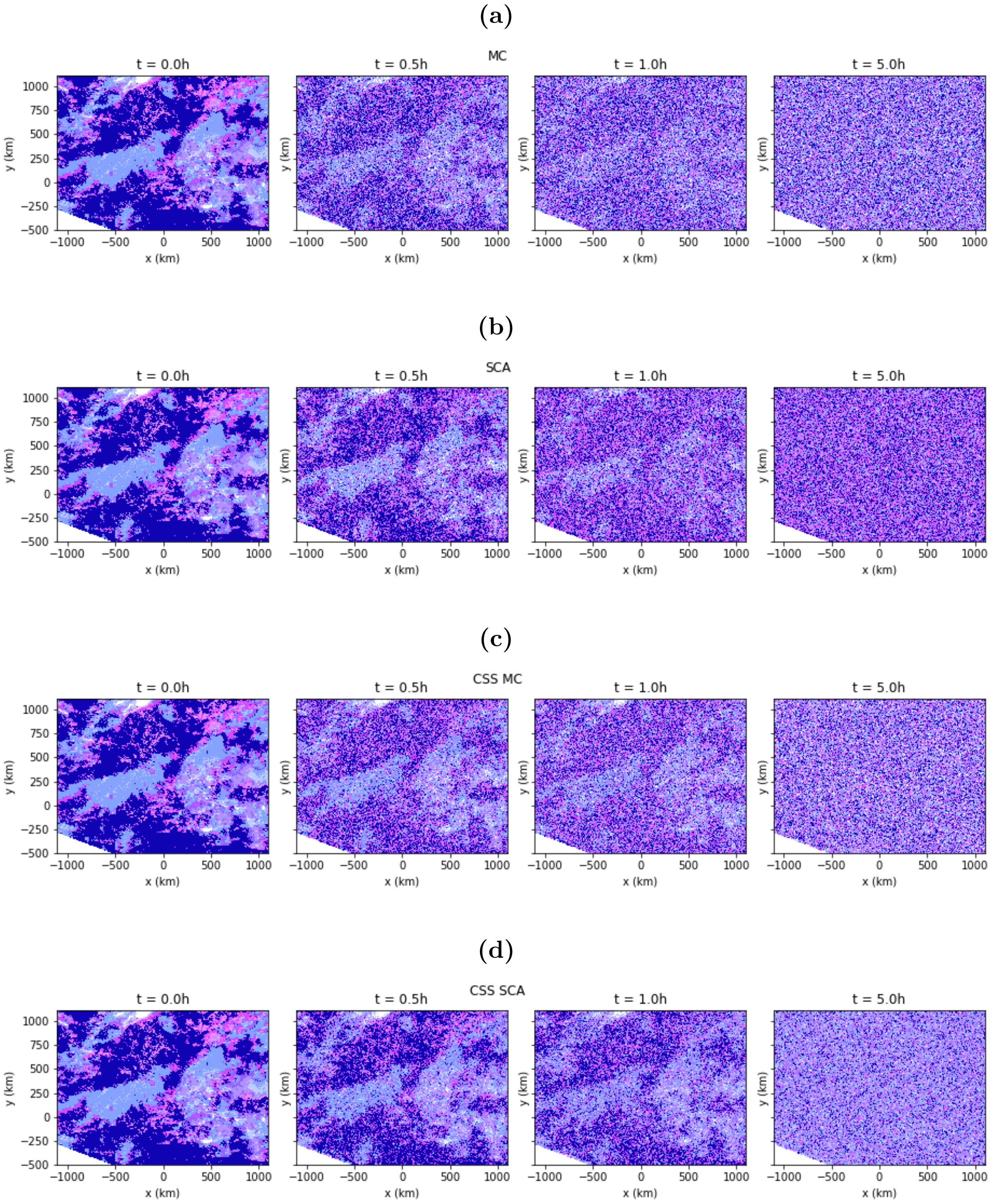
structures remain somewhat longer than in case of the discrete MC, but other than that the simulations look very similar.

At last, for the simulation of the CSS SCA, 5.2d, it can be observed that the large structures of similar cloud types remain the longest compared to the other images. However, for this model too we observe a scattered image after 5 hours.

The overall observed cloud type fractions were already shown in Table 4.1, in the **All** row. As a property of a MC the cloud type fractions of the MC converge to this overall distribution, but the other models do not necessarily. In Figure 5.3a the fractions for clear sky pixels are shown. It can be observed that the MC and the CSS MC converge to very similar values. The SCAs show a different behaviour. For the discrete SCA the clear sky fraction drops slower than the MCs, and after 1 hour the fractions go slightly up and stabilize. The CSS SCA shows a much slower decrease of clear sky pixels, but decreases to a much lower fraction than the other models.

Calculation of the organization index I_{org} is quite complicated since we have to use an algorithm to determine the different clouds as objects. Since it can be observed by eye that we are not capable to simulate the cloud structures as in the data, we do not calculate the organization index I_{org}. We did, however, calculate the standard deviation of the cloud type fractions in the macroscopic blocks $\sigma_\alpha(t)$ as we defined in Section 3.2, which was also used in Dorrestijn et al. (2013a). This metric gives an indication for the size of the spatial structures in the simulation. If this value is higher there is more variability in the clear sky fraction over the macroscopic blocks, which indicates that there are areas with more clear sky and areas with much less clear sky. So high values indicate larger structures. In Figure 5.3b, we see for all models that the standard deviation of the clear sky fraction σ_1 decreases over time, meaning that the spatial structure of the clear sky areas is becoming more regular over the domain. This is in agreement with the granulating we saw in all the simulations in Figure 5.2. It can be seen that the standard deviation for the MC and CSS MC drop fastest in the first half an hour. We saw this as well in the fast decorrelation of the panels in Figure 5.2a and 5.2c for these models. The SCAs drop somewhat slower of which the CSS SCA shows the slowest decay. This is in agreement with the longer preserved structures we observed before.

For the other cloud types similar observations for the evolution of the σ_α are obtained and are shown in Appendix C.2. The σ_α decrease over time and the CSS SCA shows the slowest decay in most cases. After about 5h of simulation, the cloud type distribution

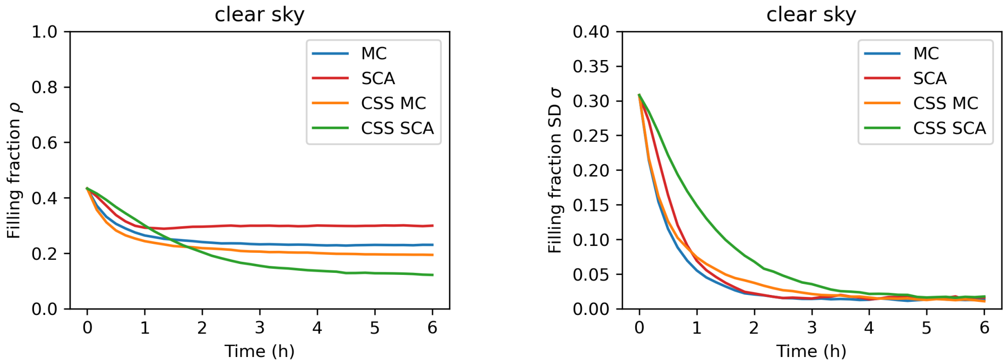

(a)

(b)

(c)

(d)

Figure 5.2: A realization for each of the four different models from chapter 4, (a) MC, (b) SCA, (c) CSS MC, (d) CSS SCA. In each figure the left panel shows the starting scene, which is one of the observations from the real data and the same for all of the models. The other three panels show the realization after 30 minutes, one hour and five hours respectively. The color coding for the cloud types is the same as in Figure 3.3b.



(a) **(b)**

Figure 5.3: The (a) clear sky fractions $\rho_1(t)$ and (b) clear sky fraction standard deviations $\sigma_1(t)$ for the four different model simulations. For the standard deviation $N = 130$ macroscopic blocks of 32×32 micro lattice sites are used.

has converged to a stationary distribution for all of the models. These are shown in Figure 5.4. By definition the cloud type distribution for the MC has converged to the overall observed cloud type occurrences from Table 4.1. It can be seen that the stationary distributions for the CSS MC is very similar to the MC, the biggest difference seems to be that there are more cumulus clouds and less clear sky. Both the CSS SCA and SCA show a stationary distribution which differs from the observed cloud type fractions.

The cloud type distributions for the CSS models can be observed in much more detail than only the cloud types and are shown in Figure 5.5. It can be seen that the overall cloud state distribution for the CSS MC are very similar to the observed cloud states in Figure 3.4. The bi-modality in the CTH is clearly observed and the log(COD) shows a bell-shaped distribution. In the data the maximum height clouds were only observed for log(COD) above 3. Here we see that in the CSS MC they are generated at any COD. Relatively, the cloud distribution of the CSS SCA shows much more mid height clouds.

5.1.2 Performance of Advection Correction

In this section we have a view at the performance of the advection correction that is used to make cloud state time series from the GOES16 images (see Section 3.1.3). The granulated images produced by the models above raise questions about the cleanness of the data as we will discuss in the Discussion (Section 6.1). In Figure 5.6 a snapshot is shown from an attached video. In this video we follow the spatial tracks of clouds at different locations. We see that the pixels follow the motion of the clouds quite well. There is an easterly wind for the lower clouds and it seems that the large high thin clouds at the left middle of the image move in the opposite direction to the north east. For one pixel we look more closely at the horizontal velocity in the x-direction and the observed cloud types over time. It can be seen that the estimated wind field is very patchy. This is due to the nearest neighbourhood estimation we used to interpolate the wind motion observations. It can also be seen in the video that the wind field changes only every hour, instead of every 10 minutes for the cloud type images, as the DMW product is available only every hour. The cloud type evolution of the explored pixel shows a very quickly

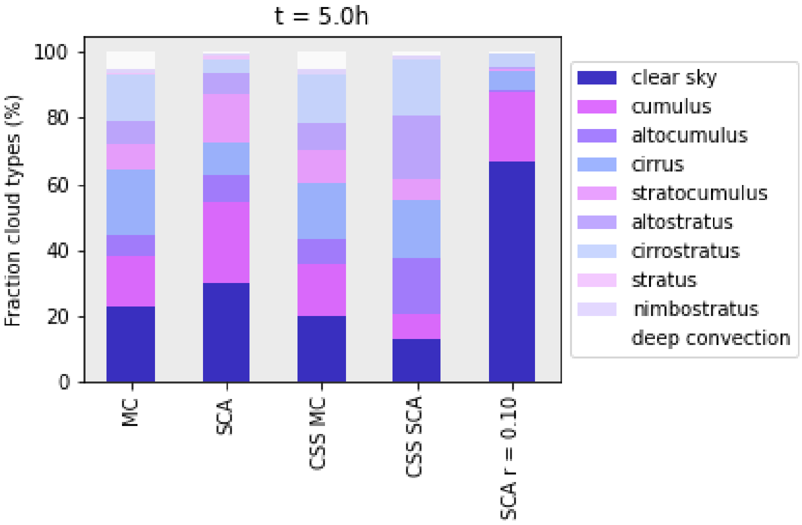
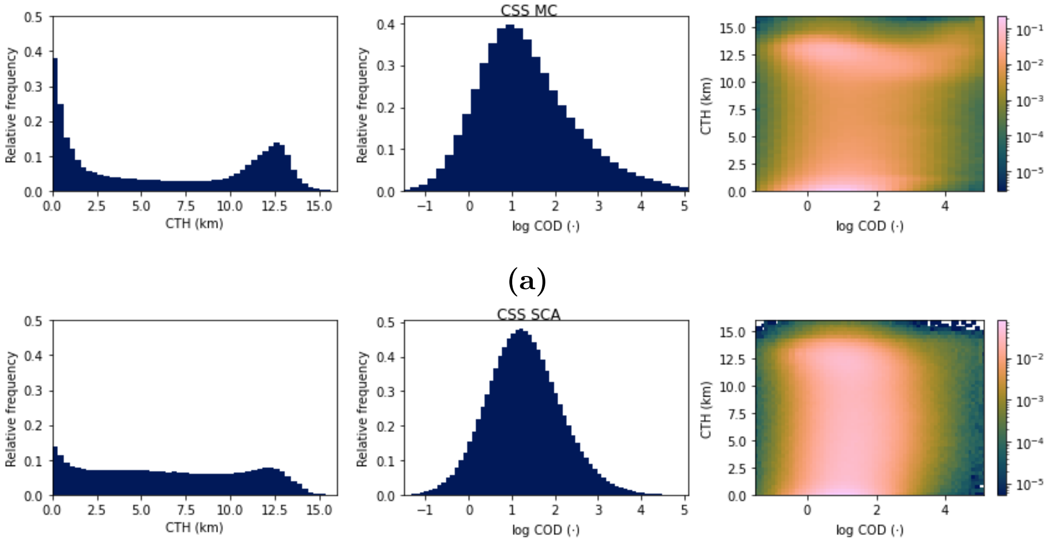


Figure 5.4: The cloud type fractions at $t = 5$ for the four different models and the SCA model with tuning parameter r as described in Section 5.2.1.



(b)

Figure 5.5: The cloud state distribution for the (a) CSS MC and (b) CSS SCA.

changing behaviour in the most right panel of Figure 5.6. It happens very often that the cloud type changes after only one time step. This is also observed for the time series of other cloud states. In the second panel it can be seen that there are many different cloud types close to one another and in the video it can be seen, that if the advection would be estimated only slightly differently the cloud state time series would change.

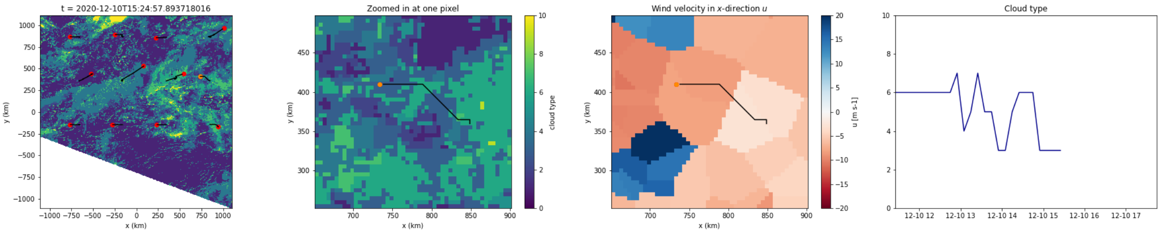


Figure 5.6: Snap shot of attached video to illustrate the advection algorithm. In the leftmost panel the cloud types at the observed domain are shown where the value corresponds with each of the cloudtypes as in Section 2.1. 12 pixels have been marked by a red dot in the initial time. For every image afterwards the estimated location of the cloud is marked by the red dot and the past track marked by the red line. In the second panel we zoomed in to the track of the one pixel that is marked with an orange dot. In the third panel the estimated wind velocity in the x-direction (East-West) is shown for the zoomed in area. In the fourth panel the observed cloud type for the orange pixel is shown over time, indicated by the corresponding numbers.

5.2 Adding tuning parameters to the models

From the results in Section 5.1 it can be seen that the larger cloud structures vanish for all of the proposed models. In this section we will show the results of a few simulations where small changes to the models have been applied, to see if it is possible that larger structures remain present over time.

5.2.1 Minimal transition probability threshold

In the resulting transition probabilities for the SCA model, we observe physically very unlikely transitions with a non-zero probability. For example the transition from cumulus to cirrostratus has a 2.5 % probability for a neighbourhood value $\gamma = 20$. We introduce a threshold parameter $r \in [0, 1]$. For transition probabilities $p_\gamma(s, k) < r$, we set $p_\gamma(s, k) = 0$. The remaining transitions probabilities are normalized such that they add up to 1 again. Of course it may happen in particular cases that for some s and γ the transition probabilities $p_\gamma(s, k) < r$ for all k. In that case we set $p_\gamma(s, s) = 1$ and $p_\gamma(s, k) = 0$ for $k \neq s$, so the pixel remains the same in the next time step with probability one.

In the extreme choices for r the following would happen. If $r = 0$, the original SCA remains since none of the transitions probabilities are smaller than zero. And if $r = 1$ all transition probabilities are set to 0. Using the exception rule the starting image will remain unchanged. In Figure 5.7 the simulation of this adjusted SCA model with tuning parameter $r = 0.10$ is shown. We see that the original structure from the starting image remains much longer than in the original simulations in Figure 5.2, e.g. the large anvil structure in the left middle of the image is still seen after five hours (the most right panel). However, the overall cloud type occurrences after 5 hours are very different from the overall cloud type occurrences, see the last column of Figure 5.4.

A similar threshold parameter $r \geq 0$ could be introduced for the CSS SCA such that for s, k, γ where the transition probability density $p_\gamma(s, k) < r$, $p_\gamma(s, k) = 0$ and

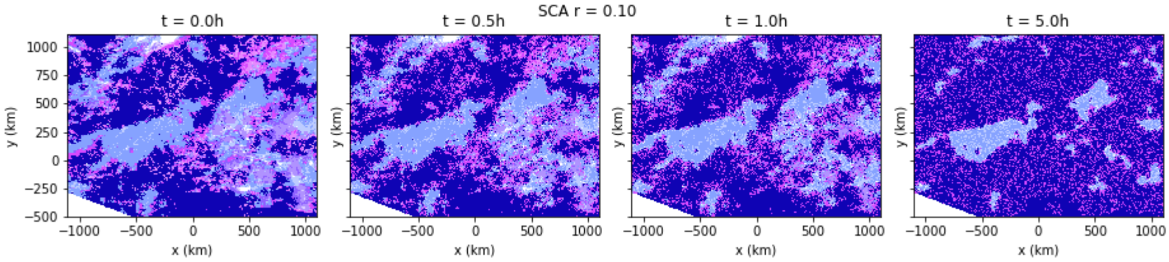


Figure 5.7: Simulation results of the SCA model with tuning parameter $r = 0.10$.

the remaining transition density is normalized again. This is not implemented, but it is expected to lead to similar results.

5.2.2 Correlating simulations of neighbouring cloud states

In all the simulations from Section 5.1, the cloud states in neighbouring locations diverge very quickly, resulting in scattered images of different cloud types. The states from one time step to the next are generated too independently. There are multiple techniques in which one can induce spatial dependence or correlation between simulated the cloud states. In this section we will try to correlate the simulated cloud states by simulating the next time step many times, and using a different variable on the grid of which we chose the correlation based on distance between different locations. This is done inspired by the method described in (Yousefi et al., 2018). The method is computationally low cost per time step, which makes it very interesting. Here we will use one specific version of the method. Using copula's this method could be implemented in different ways, but that is left for future research.

This method could be applied to both of the CSS simulations after they were trained. Here, we consider the CSS MC model and propagate only the CTH according to this method as an illustration. The time integration of the COD is just left as it was, but could easily be adjusted in a similar manner as well.

Our goal can be seen as follows. Given the realizations $S_i(t) = s_i(t)$ at some time t at all locations $i = 1, \ldots, m$, we want to simulate the following. For each i, we want to simulate the CTH $H = (H_1, \ldots, H_m) = (H_1(t + \Delta t), \ldots, H_m(t + \Delta t)$ in the following time step step where each component H_1 has marginal pdf $f_{H_i}(h)$, such that it is the marginal for the CTH from the transition density $p(s_i(t), (d, h))$ from Eq. (4.8). Next to that we also want to have some correlation between the realizations for $H_i(t + \Delta t)$ and $H_j(t + \Delta t)$ at different locations i and j.

We do this by imposing a certain Spearman rank correlation between the variables. For this we do a different simulation of an m-dimensional random variable with rank correlations corresponding to the rank correlations we want between locations. Below we will show how this is done exactly. The simulations of this random variable is done n times. By putting n realizations of the CTH H using the CSS MC, in the same order as the realizations from the extra variable, the rank correlations of the newly sorted CTHs have become as we wanted. And since we drew them from the marginals determined by the models, they also follow the right marginals. Below the exact procedure of the algorithm is written down.

So more precisely, assume that we have realizations $s_i(t)$ for some arbitrary time t at all the locations $i = 1, \dots, m$ in the grid. Simulate the CSS MC n times independently resulting in the realizations $h_i^l(t + \Delta t)$, with $l = 1, \dots n$ for the CTH at all locations. Our goal is to alter these simulations to find a simulated realization $H = h = (h_1(t + \Delta t), \dots, h_m(t + \Delta t))$ with $h_i(t + \Delta t)$ still from the marginals $f_{H_i}(h)$ and Spearman rank correlation $\rho^S(H_i, H_j)$ chosen.

Next, we will construct n realizations, $x^l = (x_1^l, \dots, x_m^l) \in \mathbb{R}^m$, $l = 1, \dots, n$, of an m-dimensional multivariate normal random variable $X = (X_1, \dots, X_m)$ with $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$, with covariance matrix $\Sigma = (\sigma_{ij}^2) \in \mathbb{R}^{m \times m}$. We have the freedom to choose the variances in the covariance matrix. Later we will see how we can choose them such that the desired rank correlations are obtained. Each of the elements of X_i of X corresponds with the location i in the grid. Let

$$\sigma_{ij}^2 = \sigma_{ref}^{2d(i,j)}, \tag{5.1}$$

where $0 < \sigma_{ref}^2 < 1$ is the reference covariance and $d(i,j)$ is the Euclidean distance between location i and j, i.e.

$$d(i,j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}, \tag{5.2}$$

where i_x, and i_y are the x- and y-index from location i respectively. So neighbouring points have distance 1, and therefore the $\mathrm{Cov}(X_i, X_j) = \sigma_{ref}^2$ for two neighbouring locations i and j. The variance of each X_i is 1., i.e. $\sigma_{ii}^2 = 1$. Due to the form of Eq. (5.1), the covariance decreases as the distance increases. Since X is a multivariate normal random variable, each X_i is also normally distributed, namely $X_i \sim \mathcal{N}(0, 1)$.

Using Cholesky decomposition we calculate the lower triangular matrix A such that

$$\Sigma = AA^T. \tag{5.3}$$

Now, we draw n independent realizations $y^l = (y_{1l}, \dots, y_{ml})$, $l = 1, \dots n$, whose m components are independently standard normal distributed to get $\widetilde{Y} = (y_{il}) \in \mathbb{R}^{m \times n}$. Then using the result of the Cholesky decomposition, let

$$\widetilde{X} = A\widetilde{Y} = (x_{il}), \tag{5.4}$$

then each column of $\widetilde{X}$, $x^l = (x_{1l}, \dots, x_{ml})$, with $l = 1, \dots, n$ is a realization of $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$.

Since each element of X has variance 1, the covariance between the components of X equals the linear Pearson correlation ρ. So

$$\rho_{ij} = \rho(X_i, X_j) = \sigma_{ij} \tag{5.5}$$

The Spearman rank correlation ρ^S between X_i and X_j is well known for multivariate normal distributions and can be expressed in terms of the linear correlation ρ (Kruskal, 1958, Eq. (6.4)), i.e.

$$\rho^S(X_i, X_j) = \frac{6}{\pi} \arcsin(\rho_{ij}/2) \frac{6}{\pi} = \arcsin(\sigma_{ij}/2). \tag{5.6}$$

Now we construct a rank matrix $R = (r_{il}) \in \mathbb{R}^{m\times n}$ which contains the ordering of the rows in $\widetilde{X}$ if we sort them. Let $x_i^{(0)} \le x_i^{(1)} \le \ldots \le x_i^{(n)}$ be the ordered elements of x_i. Then $r_{il} = k$ with k such that $x_{il} = x_i^{(k)}$. For example, for $m = 3$ locations and $n = 4$ simulations, let

$$\Sigma = AA^T = \begin{bmatrix} 1 & 0.5 & 0.2 \\ 0.5 & 1 & 0.7 \\ 0.2 & 0.7 & 1 \end{bmatrix} \tag{5.7}$$

and draw the independent standard normal distributed y_ij

$$\widetilde{Y} = \begin{bmatrix} 1.8838 & 0.8203 & 0.5402 & 0.19341 \\ -0.6985 & 0.0957 & 0.2177 & 0.19250 \\ 0.6322 & -0.0207 & -0.5435 & -0.28756 \end{bmatrix} \tag{5.8}$$

then

$$\widetilde{X} = A\widetilde{Y} = \begin{bmatrix} 1.8838 & 0.8203 & 0.5402 & 0.1934 \\ 0.3369 & 0.4930 & 0.4587 & 0.2634 \\ 0.3308 & 0.2160 & -0.1176 & -0.0271 \end{bmatrix} \tag{5.9}$$

and

$$R = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 1 & 3 & 2 & 0 \\ 3 & 2 & 0 & 1 \end{bmatrix} \tag{5.10}$$

Then we will perform a permutation on the observations in $\widetilde{H} = (h_il) \in \mathbb{R}^{m\times n}$ with $(h_i^l(t+\Delta t))$, such that the rankings are according to the matrix R, keeping the elements for one location in the same row. So for example, for fictional simulations of $\widetilde{H}$

$$\widetilde{H} = \begin{bmatrix} 9. & 5. & 3. & 6. \\ 0. & 5. & 2. & 6. \\ 0.1 & 0.3 & 0.5 & 0.7 \end{bmatrix} \tag{5.11}$$

is permutated such that we obtain,

$$H^c = \begin{bmatrix} 9. & 6. & 5. & 3. \\ 2. & 6. & 5. & 0. \\ 0.7 & 0.5 & 0.1 & 0.3 \end{bmatrix}. \tag{5.12}$$

More precisely the exact procedure to obtain the permutated observations H^c is as follows. First define a matrix with all the realizations $\widetilde{H} = (h_{il}) \in \mathbb{R}^{m\times n}$ with $h_{il} = h_i^l(t+\Delta t)$. Then sort the rows of $\widetilde{H}$ to get $H^s = (h_{il}^s)$, with $h_{il}^s = h_i(l)$, where $h_i(0) \le h_i(1) \le \ldots \le h_i(n)$ are the ordered observations at location i. Now we construct the matrix with correlated rows $H^c = (h_{il}^c) \in \mathbb{R}^{m\times n}$, by taking

$$h_{il}^c = h_{i,r_{il}}^s. \tag{5.13}$$

So we take from the simulations for location i the one realization which has the same ordering as the element x_{il} had in the i^{th} row of $\widetilde{X}$. Therefore the empirical rank correlations of $\widetilde{H}$ and $\widetilde{X}$ are the same. The rank correlations of X are in Eq. (5.6), and for large n these empirical rank correlations with be close to these theoretical rank correlations. So for n large the rank correlation of the columns of H^c will be close to the ρ^S in Eq. (5.6).

Therefore if we take for some l the realization $h^{cl} = (h^c_{1l}, \dots, h^c_{ml})$. This is a realization at each location for the state of the CTH in the next time step coming from the distributions based on the previous cloud state at each location but also with correlated values spatially. In the example, we could take for example the column $h^{c2} = (5, 5, .1)$ as final realization for H.

Since there is a direct relationship (Eq. (5.6)) between the obtained rank correlations for the CTH and the chosen covariances in Eq. (5.1), we can choose the covariances such that the desired rank correlations are obtained. These desired rank correlations could for example be inferred from the data.

In Figure 5.8 two realizations of the simulation using the method above are shown. We used $n = 10$ simulations, and take $\sigma_{ref} = 0.9$ meaning that the correlation of the between two neighbouring points equals . The complexity of the Cholesky decomposition depends on the number of locations. Therefore we decreased the size of the grid to $m = 100 \cdot 100$ gridpoints. If we would like to perform this method for a larger grid, a more advanced numerical method is needed to calculate A or directly $\widetilde{X}$. We see a big difference in the structure remained over time. Neighbouring locations have much more similar values, in the correlated simulations than when using only the CSS MC. It seems for these two simulations the number of clear sky pixels becomes much smaller than in the starting image. It is unclear whether this is something structural, and if so, why this happens.

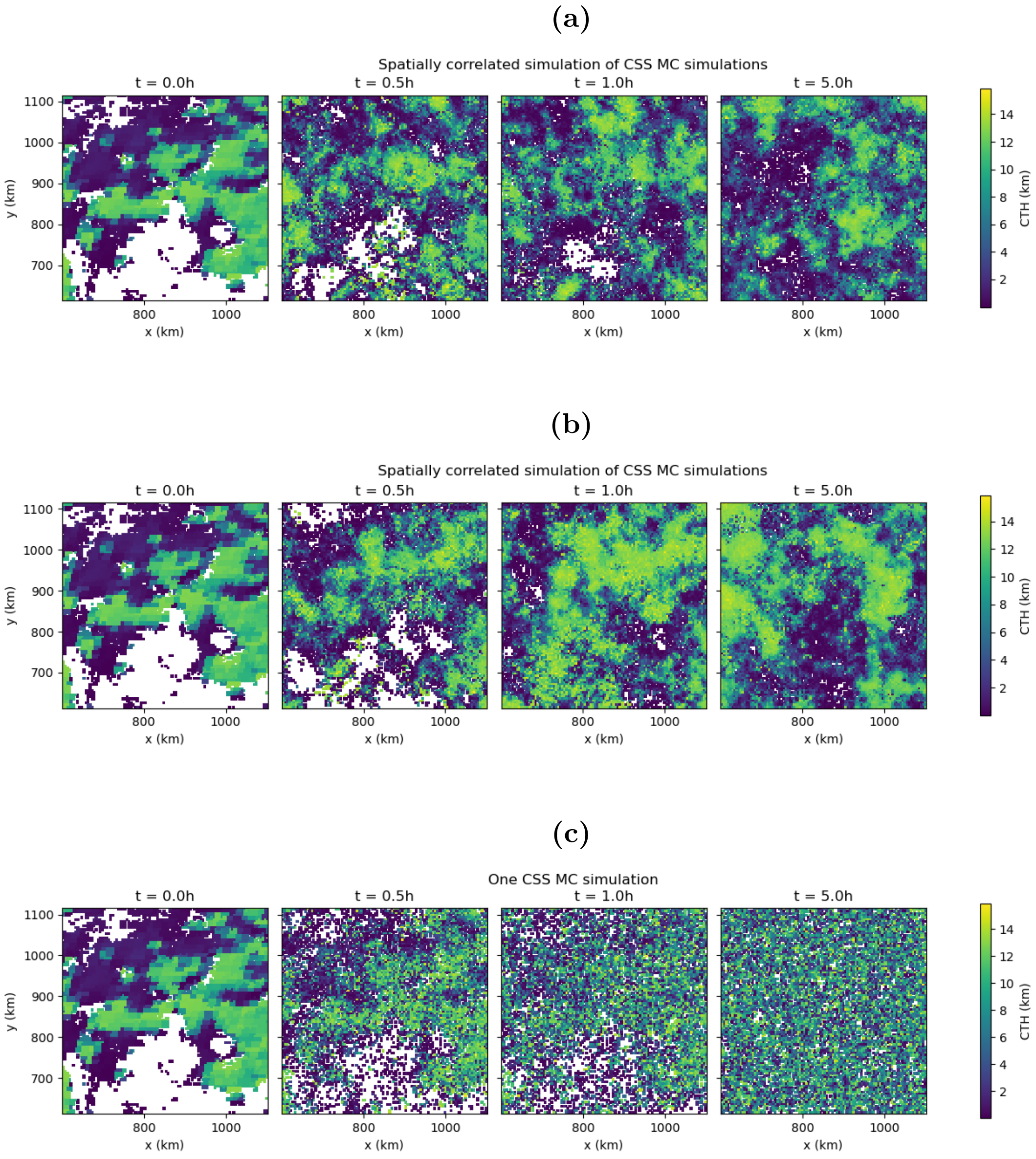


Figure 5.8: (a) and (b) Two realizations of the CTH of the CSS MC using $n = 10$ realizations every timestep to construct spatially correlated simulations. (c) One realization of the CSS MC simulation, without using the method to correlate simulations. Clear sky pixels are colored in white.

6

Discussion and Conclusion

6.1 Discussion

In this study it was tried to develop models which can simulate the spatial cloud organization as observed in the near-equator area over the Atlantic Ocean. The original models proposed in Chapter 4 were not able to capture the organization. Even in the models where spatial dependence was tried to include, the cloud states at neighbouring cells were simulated to independently. In this discussion we will explain what we observed in the original models and how more spatial dependence could be introduced in the simulations.

Each of the models simulating the clouds resulted in a granulated image after only a few hours. This develops slightly slower for the SCAs where spatial dependency is included. Although we were not able to reproduce the cloud structures we gained a lot of insight on the potential of MC models for this application.

For each of the models we observe fast changing states, such that neighbouring cells end up in very different states. For example clear sky pixels and deep convection evolve next to one another, something which is never observed in the data. This leads to neighbourhood states, which did not occur in the real data. For the SCAs this is troublesome, since the parameters are based on the neigbourhood state. In these cases the transition probabilities from the nearest observed neighbourhood state are then used as estimate for the transition probability. But these result eventually in the very unlikely granulated images.

For the MC model the cloud type filling fractions are preserved and for the CSS MC even the full cloud state in COD/CTH space is nearly preserved. However, when introducing spatial dependence the models are unable to maintain this property.

Contrary to our results the SCA from Dorrestijn et al. (2013a) showed more spatial structure. In their simulation areas of clear sky with some shallow cumulus were alternated with congestus-deep convection-stratiform mixtures. The latter mixtures were not observed in the data, but the filling fractions were still correct. In our simulations of the SCA we observe one mixture of all the different cloud types, where the filling fractions are not in agreement with the data. And the CSS SCA hardly improves either of these aspects.

There are some differences in the SCA model presented in this thesis and the SCA model in Dorrestijn et al. (2013a), which could explain the poorer performance of our SCA model and therefore also of the CSS CSA. First, the data sets on which our model and the model in Dorrestijn et al. (2013a) are trained differ. The LES data used in Dorrestijn et al. (2013a) did not contain any horizontal advection, whereas the observational data from the GOES16 satellite of course included advection. We used the Derived Motions product, provided by the GOES-16 ABI, to construct a 2d horizontal advection field. However, since this product is only available once per hour and for most areas not as detailed spatially as the COD and CTH product, we believe that a lot of pixels are mapped to completely different clouds or clear sky cells in the next time step. This leads to false observations from clouds transitioning to completely unrelated states in the next time step. When estimating the transition probabilities, or density in case of the CSS, this could lead to too high values for transitions that in reality (almost) never happened.

Another difference in the data sets is the temporal and spatial resolution, which were with 1 min and 150 m in the LES data, much higher than the 10 minutes and approximately 5 km from the GOES16 ABI images respectively. When the time step increases, the correlation between the cloud states in two subsequent time steps decreases by definition. In the models all pixels are propagated in time independently. So when neighbouring cells both change ‘too much’ in one time step, this can lead to unrealistic spatial patterns.

Finally, there are differences in the data used. The LES simulation only spanned a period of 8 hours, starting from a completely clear sky domain, which simulated the development of convective cells, where the satellite data we use in this study spans several months and includes the formation and dissipating of all cloud types. Since Dorrestijn et al. (2013a) only shows the spatial pattern of the cloud types after 8 hours of simulation, it could be that this model is only capable of simulating the organizational structure during the development of deep convection, but not for a longer period, because this was not included in the data.

The SCA model from Dorrestijn et al. (2013a) and their later models in e.g. Dorrestijn et al. (2016) do also share a property with our models which could lead to the granulating of all images. If two pixels close to one another share a very similar cloud state and neighbourhood cloud state the probability mass/density function for the next state in both pixels are very similar. This does not mean however that the two realizations for the pixels in the next time step are also very similar. In the CSS SCA we tried to improve the CSS MC by adding spatial dependence in the parameters of the transition densities, but not by including dependence between the random variables for the cloud state at adjacent locations. We suspect that a stronger coupling is needed for the evolution of clouds at pixels close to one another, such that the image in the next time step is still a likely realization spatially. One could think of different ways to include this in the models.

In Section 5.2.2 one method, based on realizations of an extra correlated variable, to include spatial correlation was implemented. This was only done for the CTH as illustration. Using the extra realizations, multiple simulations of the original models are shuffled such that the resulting CTH fields are spatially correlated. This method showed very promising results for the CTH. The CTH field showed large structures, similar to the original image, which remained over a long period of time and changed only slowly in every time step. The parameters is this method like the covariances σ_{ij} and number of simulations n could change the result of these simulations. The effect of these tuning parameters is left for future research. The method should also be implemented for the COD.

One could also think of the following. For example, if two neighbouring pixels have to probabilities p_1 and p_2 on becoming a clear sky pixel in the next time step, one could draw one realisation u from a uniform distribution on the interval [0,1]. If $u < p_1$ the first pixel becomes a clear sky pixel, and if $u > p_1$ the first pixel becomes a cloud pixel. Similarly, if $u < p_2$ the second pixel becomes a clear sky pixel and if $u > p_2$ it becomes a cloud pixel. If u is very small, both pixels become clear sky and if u is very large they will both become a cloud pixel. Something similar could be done for drawing from the cloud transition distribution $f^{(c)}$ using *inverse transformation sampling.* Instead of one single realization u, one could also think of a spatial realization using a Gaussian field sample.

Another method that could be researched, is including spatial transition distributions. In the same way a pixel transitions from one cloud state to another state over time, one could think of a transition from the state in a pixel to the state in a neighbouring pixel. If we also estimate the spatial transition distributions, we could draw a realization of the next time step based on both the temporal and spatial transition distribution. This could be done using for example Gibbs-sampling or sequential Gaussian simulation.

Finally, we also propose a method where we consider the clouds as objects. To prevent clouds to be destroyed from within we could also add a constraint the state at a pixel can only change if this pixel is at the edge of a cloud. One could also think if adding different constraints per cloud type. For example that cirrus is a little more likely to change into clear sky from within the cloud, where as this would never happen for deep convective clouds.

Some model choices could also be reconsidered to come with more elaborate methods. For example the choice for the Nadaraya-Watson estimator could be improved by taking a different kernel and making the kernel bandwidth adaptive as in, e.g., Demir and Toktamiş (2010). Also the construction of the cloud transition function could be chosen differently, with for example, taking dependence between the next state cloud top height and cloud optical depth into consideration through a copula. Also instead of taking a fixed maximum CTH h_{max} this parameter could also be estimated as a function of the COD. This could solve the problam that only clouds with high COD can penetrate through the Tropopause resulting in much higher CTH.

Also the estimation of the mutual information between neighbourhood variables and the next cloud state could be improved by estimating the densities with kernel estimators or using adaptive bandwidths, e.g. like in Hu et al. (2011).

It should also be noted that the models are time and space invariant, and we tried to use data which has this property. We did this by taking an area over sea, such that the terrain is always similar and the surface heat and moisture fluxes are more constant over time, since sea heats much slower than land. If the parameterizations based on these models were to be used in GCMs these dependencies should also be researched.

One other post-processing manner was found to simulate spatial structures, by introducing a tuning parameter r. Setting transition probabilities which were smaller than r to zero led to large cloud structures for a much longer time period than all the original models proposed in this study. By reducing the estimated unlikely cloud type transitions, some of the false observations due to wrongly estimated advection are removed from the estimated data, but maybe also some unlikely transitions that do occur in reality. In any case it shows that a model which has less states to transition to with positive probability shows more conservation of cloud structures. This result encourages the improvement of advection correction, since it could be the solution to less randomly changing states.

In this study we try to improve the cloud type models from Dorrestijn et al. (2013a) by introducing the cloud states in a continuous state space. This showed promising results regarding the conservation of the cloud state distributions, while we impose parametric PDFs on the next state distributions. This could help in constructing a more precise parameterization in large scale models for the sub grid processes. Whether this improves the prognostic performance of the large scale models is left for future research.

6.2 Conclusion

The aim of this thesis was to improve the capability of Markov Chain models to simulate the dynamics of spatial organization of cloud types using observational data. These models can be used for the construction of convection parameterization in global weather and climate models. Two additions to the original models in Dorrestijn et al. (2013a) have been proposed; (1) describing the state of the clouds in a continuous state space using the COD and CTH, instead of a finite number of cloud types and (2) a search for better aggregation functions of the neighbourhood of the cell which influence the transition probabilities (densities in case of the continuous state space) to the next cloud state. The models were trained using the GOES16 satellite images for the cloud top height and cloud optical depth.

It was found that the CSS MC was capable to reproduce the overall cloud state distributions as observed, but off course no spatial structures could be reproduced since there was no spatial coupling in the model. After that we tried to add spatial dependence to both the MC and CSS MC with the SCA and CSS SCA respectively, to simulate spatial cloud organization. However, we were not able to reproduce the cloud organization with our SCA as was simulated by the SCA in Dorrestijn et al. (2013a) which used Large Eddy Simulation training data. Using a CSS SCA did not improve the simulating of organizational structures. This was most likely due to the 'spatial dependence' introduced in the SCA models. Only the distributions for the cloud states in neighbouring cells were similar, but the next states were still simulated independently. Using multiple simulations per time step we were able to construct realization with spatial correlation between locations nearby. This or other methods should be researched to simulate dependent variables from known marginals. Also, by adding spatial dependence to the MC models, the property of producing the right cloud state occurrence frequencies was not a given anymore.

Artificially we tried to mimic the effect of reducing the noise on the advection correction by introducing a transition probability threshold parameter r. This resulted in much longer preservation of spatial structures, although still not representative of observed spatial cloud structures. Therefore we believe the models should be tested on a data set where a better advection correction algorithm is used to determine the location of a cloud in the next image.

Further more it should be noticed that these models only use observations over sea near the equator with no variation spatially. When introducing parameterizations using the MC models for modelling clouds over other type of surfaces this should be taken into account.

Dorrestijn et al. (2015) and Dorrestijn et al. (2016) tried to improve the MC models from Dorrestijn et al. (2013a) by conditioning on the large scale variables. In this study we focused on modelling the MC in a continuous state space and the direct dependence of neighbouring cells. Using observational data from a relative large area no better conditioning of neighbouring cells was found, but the continuous state space was well possible and can be tested for parameterizations in models.

For further research in the spatial MC models, we recommend to implement models which contain a stronger coupling between the cloud states in neighbouring pixels. As stated in the discussion, this could for example be done using some extra variable which contains correlation as described in section 5.2.2, a Gaussian field and applying inverse transformation sampling or by defining the spatial transition densities and sampling using for example Gibbs sampling. The first method showed very promising results, and it would be interesting to test the effect of the parameters in the method. For example, the spatial correlation could be inferred from the observed images. Also a better advection correction should be explored.

References

Alexandrov, M. D. and Marshak, A. (2017). “Cellular statistical models of broken cloud fields. Part III: Markovian properties”. *J. Atmos. Sci.*, 74(9), pp. 2921–2935 (cit. on p. 1).

Battiti, R. (1994). “Using mutual information for selecting features in supervised neural net learning”. *IEEE Transactions on Neural Networks*, 5(4), pp. 537–550 (cit. on p. 33).

Bengtsson, L., Steinheimer, M., Bechtold, P., and Geleyn, J. F. (2013). “A stochastic parametrization for deep convection using cellular automata”. *Q. J. R. Meteorol. Soc.*, 139(675), pp. 1533–1543 (cit. on p. 1).

Böing, S. (2016). “An object-based model for convective cold pool dynamics”. *Math. Clim. Weather Forecast.*, 2(1), pp. 43–60 (cit. on p. 17).

Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). “Conditional likelihood maximisation: a unifying framework for information theoretic feature selection”. *The journal of machine learning research*, 13(1), pp. 27–66 (cit. on p. 33).

Demir, S. and Toktamiş, Ö. (2010). “ON THE ADAPTIVE NADARAYA-WATSON KERNEL REGRESSION ESTIMATORS”. *Hacettepe Journal of Mathematics and Statistics*, 39(3), pp. 429–437 (cit. on p. 53).

Daniels, J., Bresky, W., Wanzong, S., Velden, C., and Berger, H. (2019). *GOES-R Advanced Baseline Imager (ABI): Algorithm Theoretical Basis Document For Derived Motion Winds.* available at: `https://www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_Winds_v3.1_Feb2019.pdf`. Last accessed: 2022-05-20 (cit. on p. 15).

Darbellay, G. A. (1999). “An estimator of the mutual information based on a criterion for conditional independence”. *Computational Statistics & Data Analysis*, 32(1), pp. 1–17 (cit. on p. 33).

Doquire, G. and Verleysen, M. (2013). “Mutual information-based feature selection for multilabel classification”. *Neurocomputing*, 122. Advances in cognitive and ubiquitous computing, pp. 148–155 (cit. on p. 32).

Dorrestijn, J., Crommelin, D. T., Biello, J. A., and Böing, S. J. (2013a). “A data-driven multi-cloud model for stochastic parametrization of deep convection”. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, 371(1991) (cit. on pp. 1, 2, 5–8, 16, 19–22, 32, 40, 51, 52, 54, 55).

Dorrestijn, J. (2016). “Stochastic convection parameterization”. PhD thesis. Delft University of Technology (cit. on p. 2).

Dorrestijn, J., Crommelin, D. T., Siebesma, A. P., and Jonker, H. J. (2013b). “Stochastic parameterization of shallow cumulus convection estimated from high-resolution model data”. *Theor. Comput. Fluid Dyn.*, 27(1-2), pp. 133–148 (cit. on p. 1).

Dorrestijn, J., Crommelin, D. T., Siebesma, A. P., Jonker, H. J., and Jakob, C. (2015). “Stochastic parameterization of convective area fractions with a multicloud model inferred from observational data”. *J. Atmos. Sci.*, 72(2), pp. 854–869 (cit. on pp. 2, 55).

Dorrestijn, J., Crommelin, D. T., Siebesma, A. P., Jonker, H. J., and Selten, F. (2016). “Stochastic convection parameterization with Markov chains in an intermediate-complexity GCM”. *J. Atmos. Sci.*, 73(3), pp. 1367–1382 (cit. on pp. 2, 52, 55).

Forster, P., Storelvmo, T., Armour, K., Collins, W., Dufresne, J.-L., Frame, D., Lunt, D., Mauritsen, T., Palmer, M., Watanabe, M., Wild, M., and Zhang, H. (2021). “The Earth’s Energy Budget, Climate Feedbacks, and Climate Sensitivity. In Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change”, pp. 923–1054 (cit. on p. 1).

Hagos, S., Feng, Z., Plant, R. S., Houze Jr., R. A., and Xiao, H. (2018). “A Stochastic Framework for Modeling the Population Dynamics of Convective Clouds”. *Journal of Advances in Modeling Earth Systems*, 10(2), pp. 448–465 (cit. on p. 1).

Hall, M. A. (2000). “Correlation-based feature selection of discrete and numeric class machine learning” (cit. on p. 33).

Heidinger, A. (2012). *GOES-R Advanced Baseline Imager (ABI): ABI Cloud Height*. available at: `https://www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_Cloud%20Height_v3.0_July%202012.pdf`. Last accessed: 2022-05-20 (cit. on p. 13).

Hu, Q., Zhang, L., Zhang, D., Pan, W., An, S., and Pedrycz, W. (2011). “Measuring relevance between discrete and continuous features based on neighborhood mutual information”. *Expert Systems with Applications*, 38(9), pp. 10737–10750 (cit. on p. 53).

Janssens, M., Vilà-Guerau de Arellano, J., Scheffer, M., Antonissen, C., Siebesma, A. P., and Glassmeier, F. (2021). “Cloud Patterns in the Trades Have Four Interpretable Dimensions”. *Geophys. Res. Lett.*, 48(5) (cit. on p. 16).

Khouider, B., Biello, J., and Majda, A. J. (2010). “A stochastic multicloud model for tropical convection”. *Commun. Math. Sci.*, 8(1), pp. 187–216 (cit. on pp. 1, 5, 6).

Kruschke, J. K. (2015). “Chapter 6 - Inferring a Binomial Probability via Exact Mathematical Analysis”. In: *Doing Bayesian Data Analysis (Second Edition)*. Ed. by J. K. Kruschke. Second Edition. Boston: Academic Press, pp. 123–141 (cit. on p. 62).

Kruskal, W. H. (1958). “Ordinal Measures of Association”. *Journal of the American Statistical Association*, 53(284), pp. 814–861 (cit. on p. 46).

Minnis, P. and Heck, P. W. (2012). *GOES-R Advanced Baseline Imager (ABI): Algorithm Theoretical Basis Document For Nighttime Cloud Optical Depth, Cloud Particle Size, Cloud Ice Water Path, and Cloud Liquid Water Path (NCOMP)*. available at: `https:`

//www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_Cloud_NCOMP_v3.0_Jul2012.pdf. Last accessed: 2022-06-10 (cit. on p. 12).

Moddemeijer, R. (1989). "On estimation of entropy and mutual information of continuous distributions". *Signal Processing*, 16(3), pp. 233–248 (cit. on p. 33).

Ormes, J. (2017). "Cosmic Rays and Climate". *Advances in Space Research*, 62 (cit. on p. 7).

Prigarin, S. M. and Marshak, A. (2009). "A simple stochastic model for generating broken cloud optical depth and cloud-top height fields". *J. Atmos. Sci.*, 66(1), pp. 92–104 (cit. on p. 1).

Sakradzija, M., Seifert, A., and Dipankar, A. (2016). "A stochastic scale-aware parameterization of shallow cumulus convection across the convective gray zone". *Journal of Advances in Modeling Earth Systems*, 8(2), pp. 786–812 (cit. on p. 1).

Shannon, C. E. (1948). "A mathematical theory of communication". *The Bell system technical journal*, 27(3), pp. 379–423 (cit. on p. 33).

Tibshirani, R. (1996). "Regression Shrinkage and Selection Via the Lasso". *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), pp. 267–288 (cit. on p. 33).

Tompkins, A. M. and Semie, A. G. (2017). "Organization of tropical convection in low vertical wind shears: Role of updraft entrainment". *Journal of Advances in Modeling Earth Systems*, 9 (2), pp. 1046–1068 (cit. on p. 17).

Tsimpiris, A., Vlachos, I., and Kugiumtzis, D. (2012). "Nearest neighbor estimate of conditional mutual information in feature selection". *Expert Systems with Applications*, 39(16), pp. 12697–12708 (cit. on p. 33).

Varin, C., Reid, N., and Firth, D. (2011). "An overview of composite likelihood methods". *Statistica Sinica*, pp. 5–42 (cit. on p. 27).

Vergara, J. R. and Estévez, P. A. (2014). "A review of feature selection methods based on mutual information". *Neural Comput. Appl.*, 24(1), pp. 175–186 (cit. on p. 34).

Walters-Williams, J. and Li, Y. (2009). "Estimation of Mutual Information: A Survey". In: *Rough Sets and Knowledge Technology*. Ed. by P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 389–396 (cit. on p. 33).

Walther, A., Straka, W., and Heidinger, A. (2013). *GOES-R Advanced Baseline Imager (ABI): Algorithm Theoretical Basis Document For Daytime Cloud Optical and Microphysical Properties (DCOMP)*. available at: https://www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_Cloud_DCOMP_v3.0_Jun2013.pdf. Last accessed: 2022-05-20 (cit. on p. 12).

Wand, M. P. and Jones, M. C. (1994). *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability 60. Boca Raton, FL, U.S.: Chapman & Hall (cit. on p. 23).

Weger, R., Lee, J., Zhu, T., and Welch, R. (1992). “Clustering, randomness and regularity in cloud fields: 1. Theoretical considerations”. *Journal of Geophysical Research: Atmospheres*, 97(D18), pp. 20519–20536 (cit. on pp. 16, 17).

Yousefi, B., Ajina, M., Imran, M., and Laskey, K. (2018). “A NON-PARAMETRIC APPROACH TO SIMULATE PANEL DATA”. In: *2018 Winter Simulation Conference (WSC)*, pp. 515–524 (cit. on p. 45).

A

Generic Mathematical Concepts

A.1 Beta mixture distribution

If a random variable X is Beta distributed, its probability density f_X is characterized by the two shape parameters, $\alpha, \beta > 0$. It is defined for $0 < x < 1$ by

$$f_X(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)}(1-x)^{\alpha-1}x^{\beta-1}, \tag{A.1}$$

where $B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\mathrm{d}t$, also known as the *Beta function*, is a normalizing constant, such that the integral over the distribution equals 1. Beta distributed variables can only take values within the bounded domain $[0,1]$. In Figure A.1 some shapes of the distribution are shown for different values of α and β. If both values are equal to 1 the distribution simplifies to the uniform distribution on the interval $[0,1]$. For $\alpha = \beta$ the distribution is symmetric around $x = 0.5$.

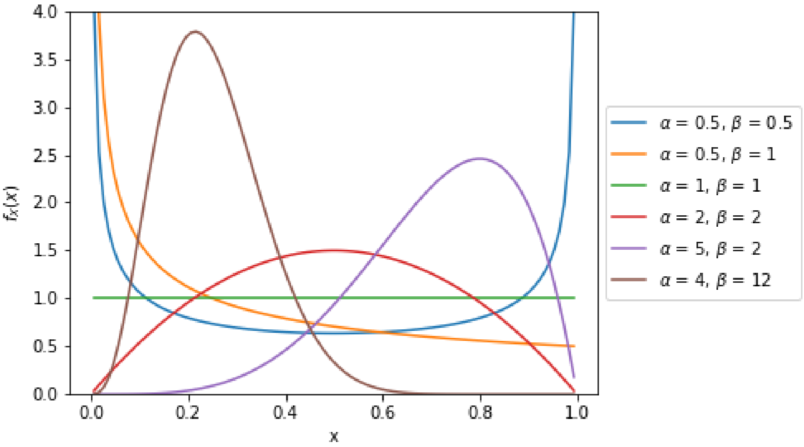


Figure A.1: Beta probability density function for different values of α and β.

A.1.1 Different parameterization

The shape parameters α and β are hard to intuitively interpret. There are different combinations of parameters possible which define the Beta distribution. It can also be reparameterized by the much more intuitive mean $\mu = \frac{\alpha}{\alpha+\beta}$ and the sum of the shape parameters $\nu = \alpha + \beta$, which is a measure for the 'concentration' of the distribution, see Kruschke (2015). As ν gets larger the distribution becomes narrower. Since these parameters give a better intuition for the shape of the distribution these are further used in the analysis. In this thesis when a random variable $X \sim \mathrm{Beta}(\mu, \nu)$ the reparameterized distribution is meant, with probability density function

$$f_X(x; \mu, \nu) = \frac{1}{\mathrm{B}(\mu\nu, \nu(1-\mu))} x^{\mu\nu-1}(1-x)^{\nu(1-\mu)-1} = \frac{1}{\mathrm{B}(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}.$$

A.1.2 Parameter estimation methods

When estimating the parameters of a Beta distribution a closed form is not existing for the maximum likelihood (ML) estimator. The method of moment (MoM) estimator however does have the closed form, namely

$$\hat{\mu}^{MoM} = m_1, \tag{A.2}$$

$$\hat{\nu}^{MoM} = \begin{cases} \frac{m_2 - m_1}{m1^2 - m_2}, & \text{if } m_1^2 > m_2 \\ \frac{m_1 - m_2}{m1^2 - m_2}, & \text{if } m_1^2 < m_2, \end{cases} \tag{A.3}$$

where $m_j = \sum_i^n x_i^j$ is the j-th sample moment. This MoM estimator can be a good intial estimate for an ML estimation algorithm.

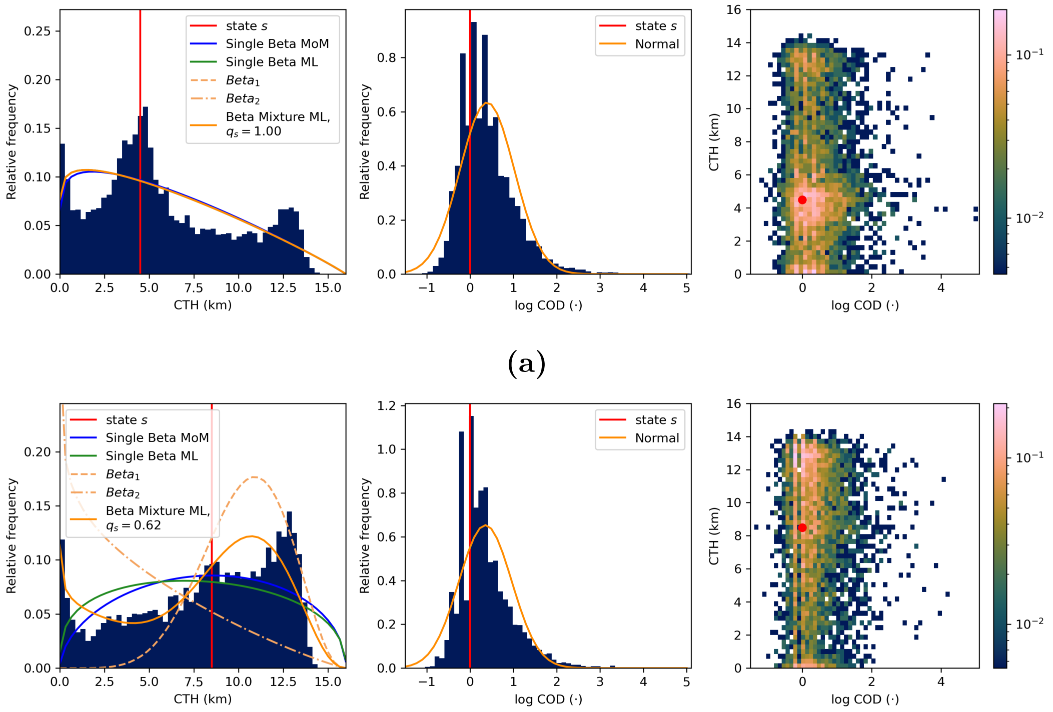
In this thesis we also use a mixture of two Beta distributions. For this mixture a closed form is not existing for both the ML and MoM estimator. Therefore we will use the Nelder-Mead algorithm with an educated guess as initial estimates for the estimation of the ML estimator.

B

Additional figures for model development

B.1 Histograms of next time step cloud states

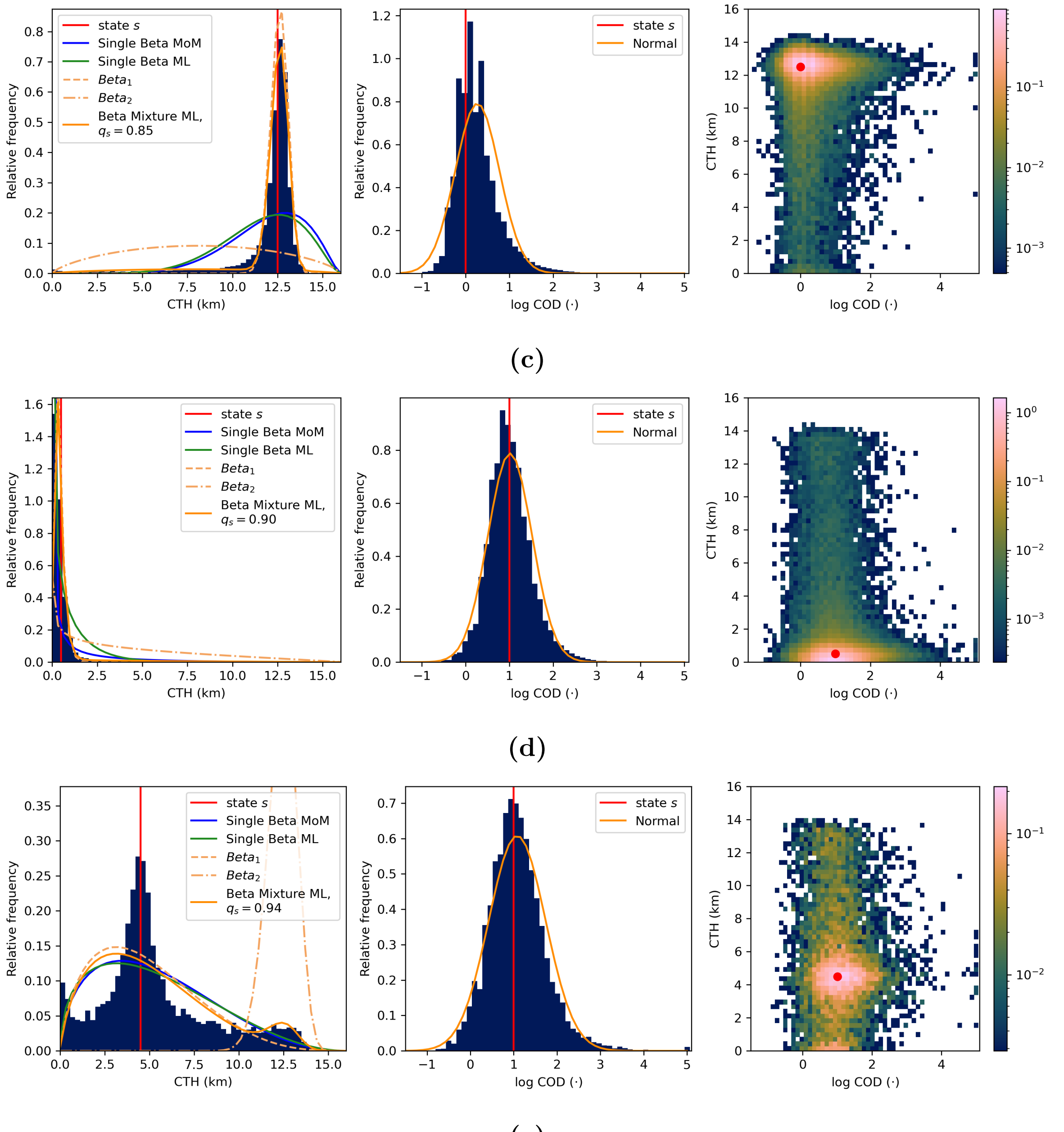
Given observed cloud states ‘close’ to a state s, the observed distributions of the subsequent time step states are shown in this appendix. The histograms for a number of choices of the starting state s are shown in each subplot in Figure B.1.



(a)

(b)

Figure B.1: Histograms of states $s_i(t + \Delta t) \in \Omega_c$, with i, t such that the previous state $s_i(t)$ is close to s, i.e. $K_{(\Delta d, \Delta h)}(s - s_i(t)) = 1$ with $\Delta h = 200$m and $\Delta d = 0.2$. The value of s is indicated by the red lines and dot. The fitted distributions as described in Section 4.3 are also shown.

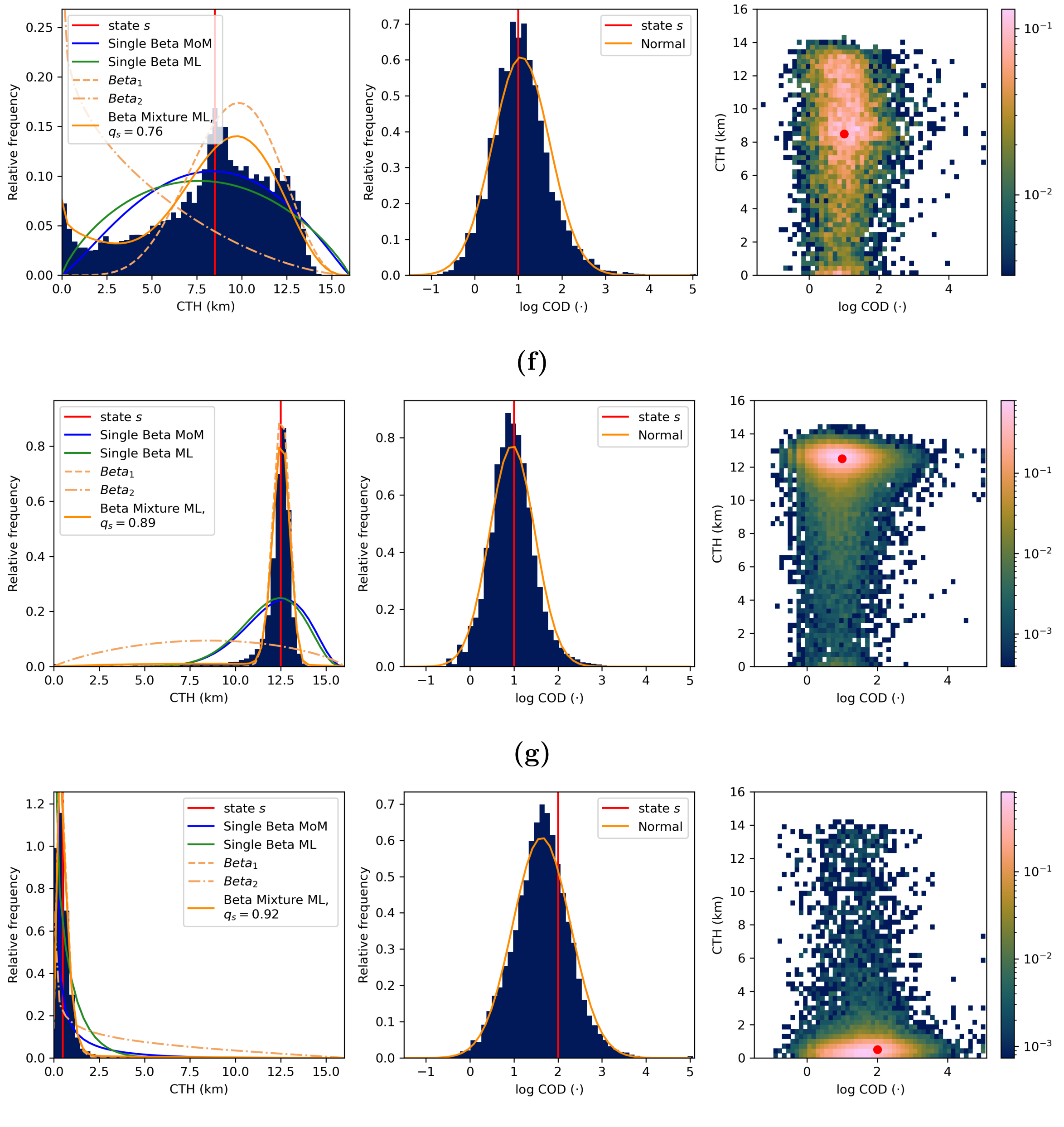


(c)

(d)

(e)

Figure B.1: (Cont.)

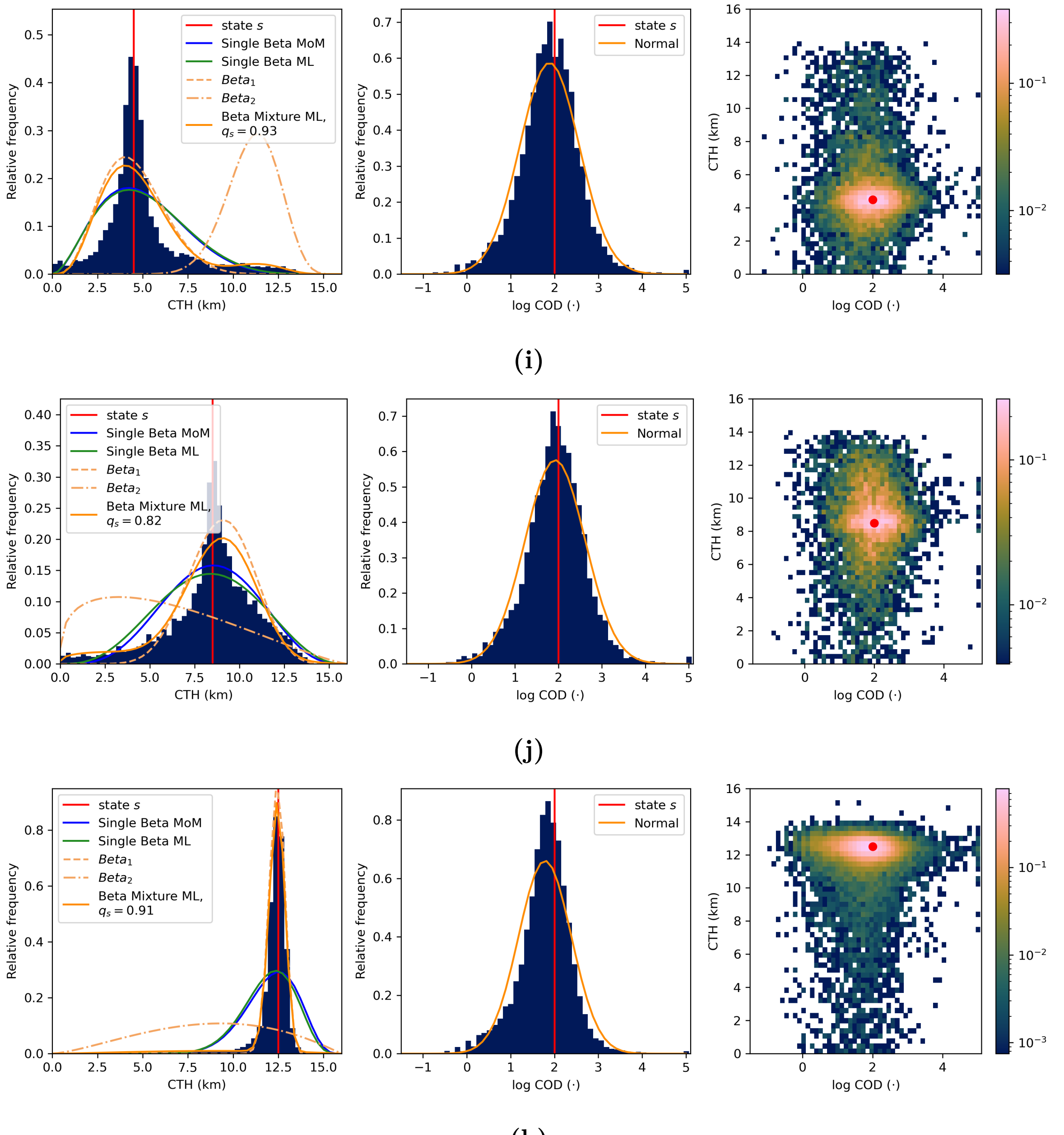


(f)

(g)

(h)

Figure B.1: (Cont.)

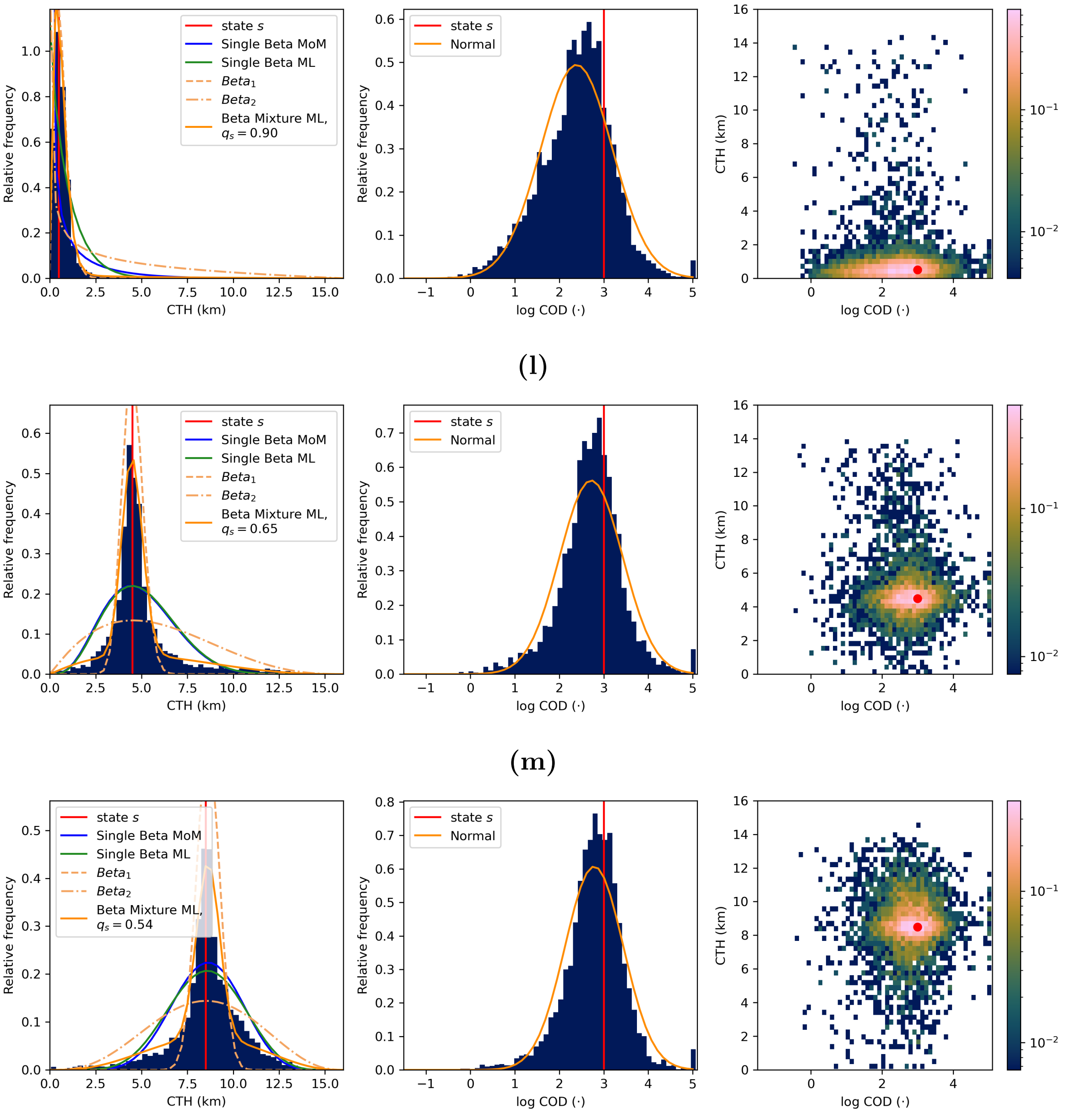


(i)

(j)

(k)

Figure B.1: (Cont.)



(l)

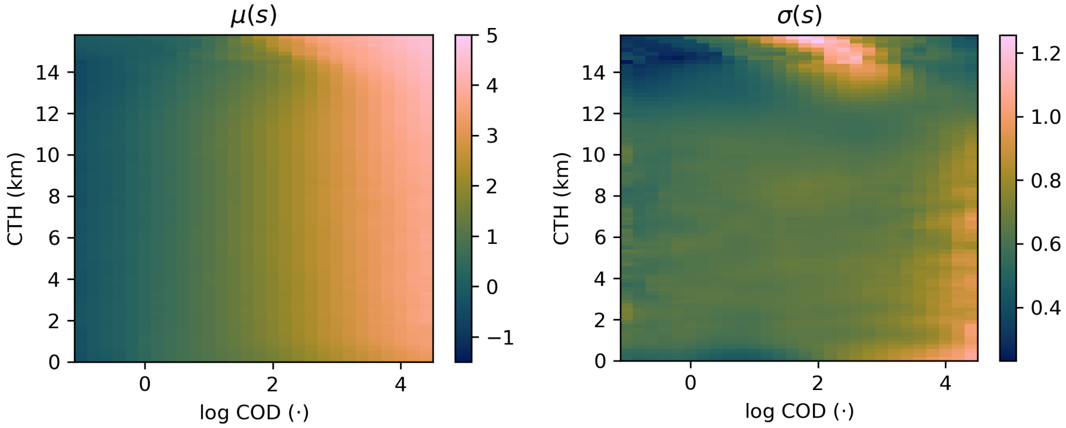
(m)

(n)

Figure B.1: (Cont.)

B.2 Estimates of distribution parameters for CSS MC

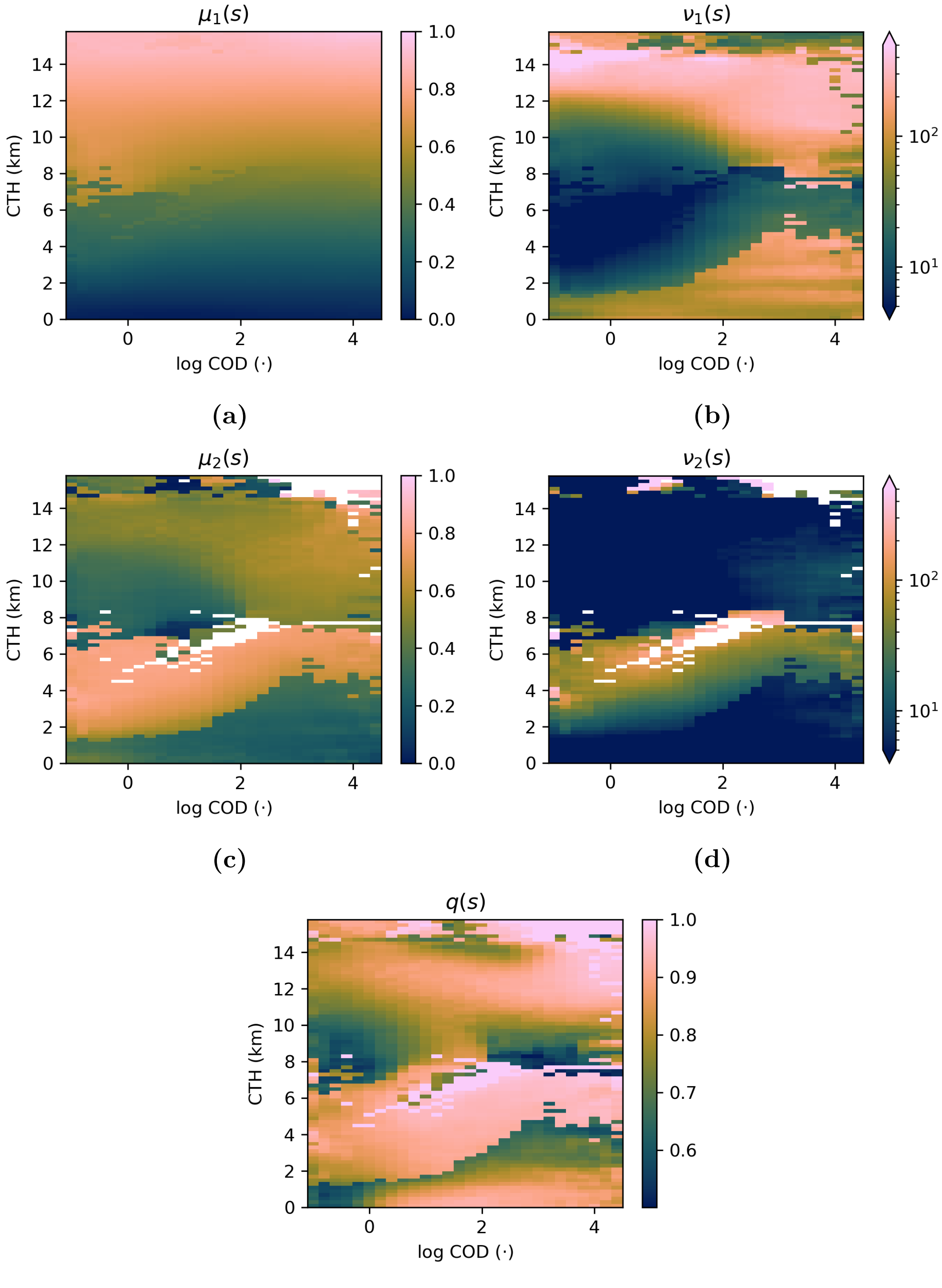
The estimates of the parameters for the normal and Beta-mixture distributions from equations (4.29) and (4.30), for all estimated starting states $s = (d, h)$.



(a)

(b)

Figure B.2: Estimates for the parameters from Eq. (4.29) with bin widths $\Delta h = 200\text{m}$ and $\Delta d = 0.2$.



(a)

(b)

(c)

(d)

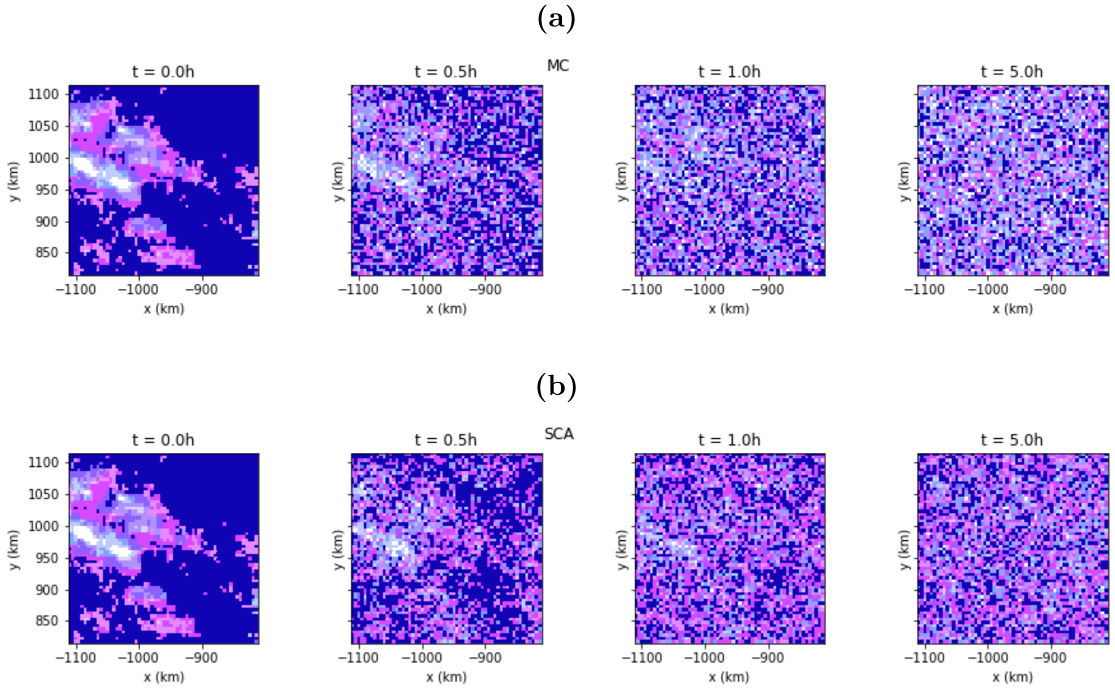
(e)

Figure B.3: Estimates for the variables from Eq. (4.30) with bin widths $\Delta h = 200$m and $\Delta d = 0.2$.

C

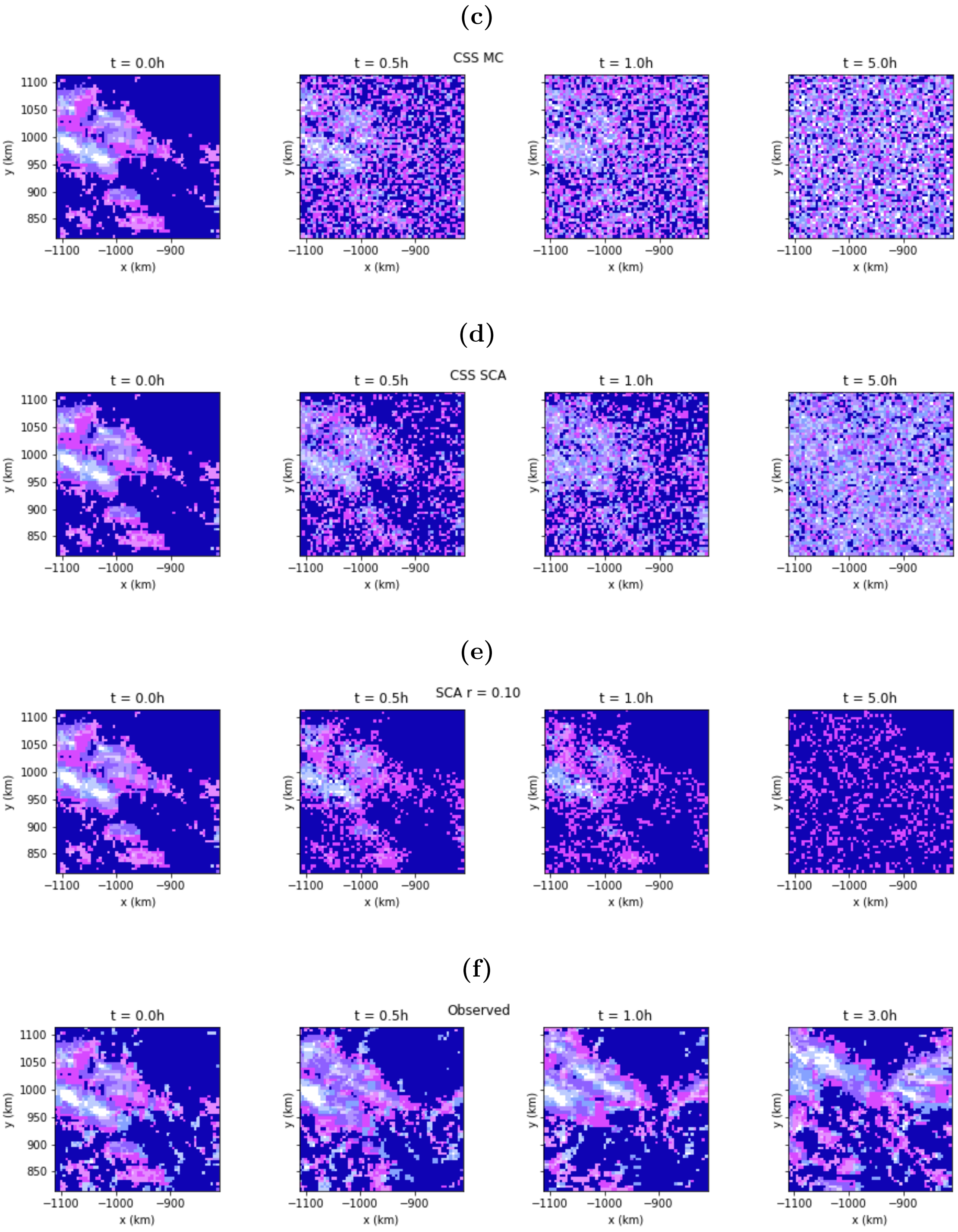
Additional figures of simulations

C.1 Zoomed-in images of the realizations for the different models


(a)

(b)

Figure C.1: Zoomed-in images of the realizations from Figure 5.2 for each of the four different models from chapter 4, (a) MC, (b) SCA, (c) CSS MC, (d) CSS SCA, (e) SCA with $r = 0.10$ and (f) the observed images. In each figure the left panel shows the starting scene, which is one of the observations from the real data and the same for all of the models. The other three panels show the realization after 30 minutes, one hour and five hours respectively. The color coding for the cloud types is the same as in Figure 3.3b.



(c)

(d)

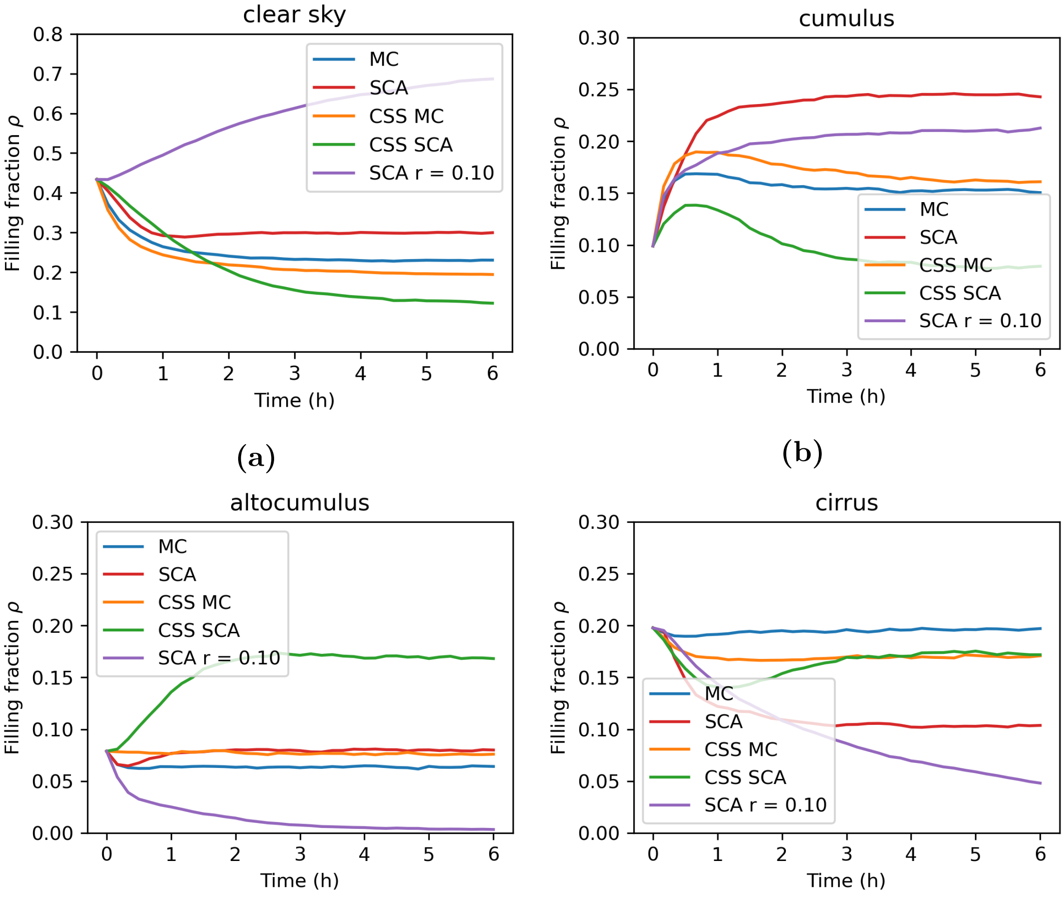
(e)

(f)

Figure C.1: (Cont.)

C.2 Cloud type filling fractions

In this appendix the evolution of the cloud type filling fractions are shown for each cloud type. Remark that the extend of the y-axis is different for clear sky.



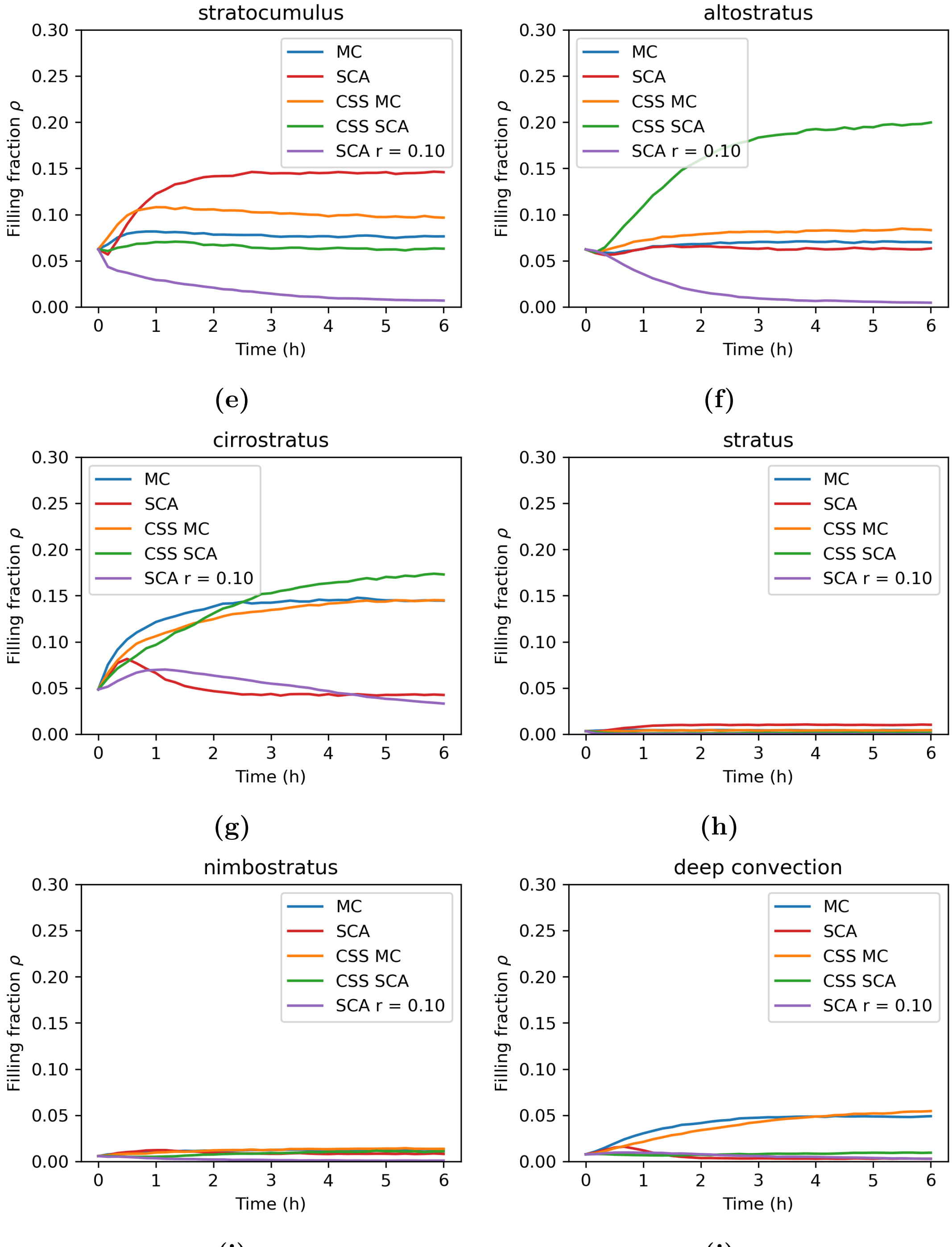
(a)

(b)

(c)

(d)

Figure C.2: Cloud type filling fractions $\rho_\alpha(t)$ for all cloud types α in the different subfigures. The fractions are shown over time for all the different models proposed in the thesis.



(e)

(f)

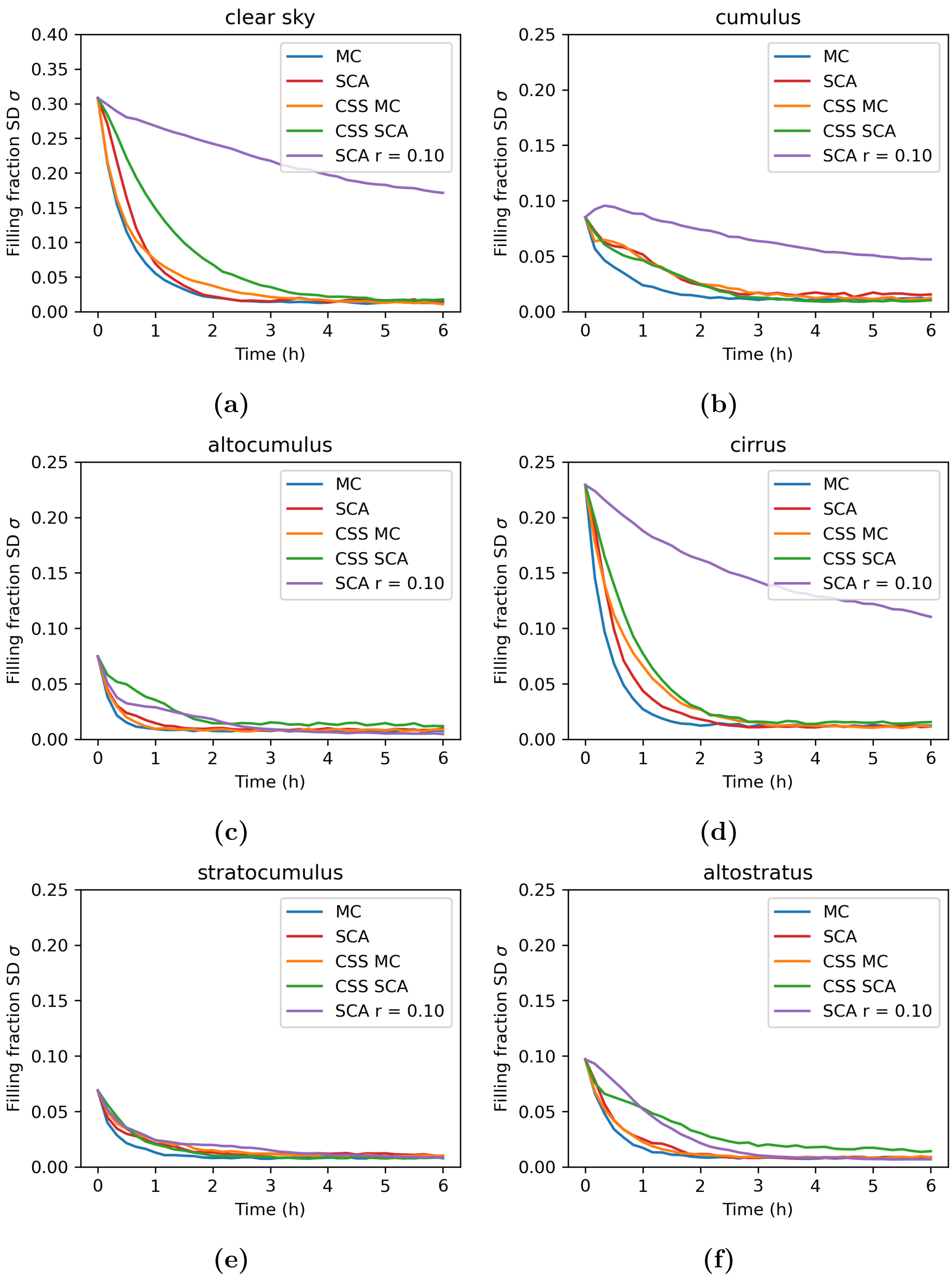
(g)

(h)

(i)

(j)

Figure C.2: Cloud type filling fractions $\rho_\alpha(t)$ for all cloud types α in the different subfigures. The fractions are shown over time for all the different models proposed in the thesis. (Cont.)



(a)

(b)

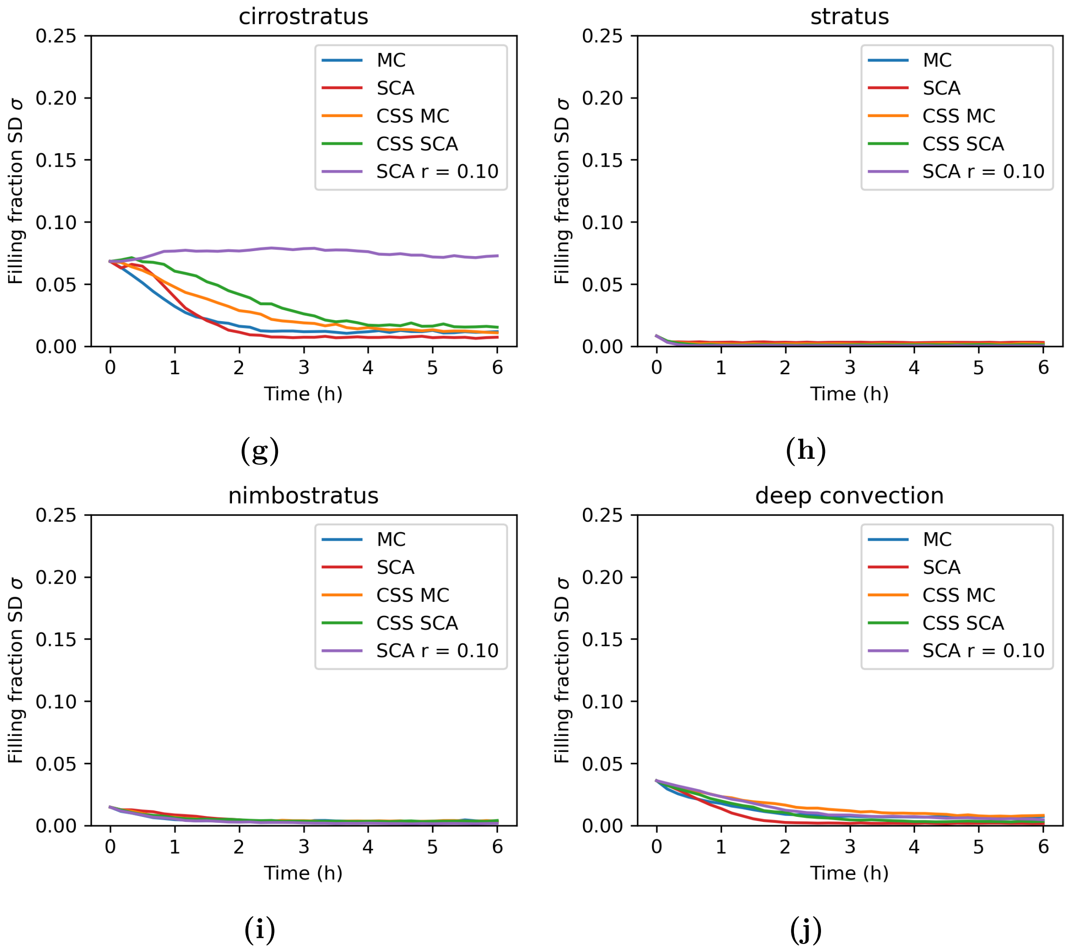
(c)

(d)

(e)

(f)

Figure C.3: Cloud type filling fractions standard deviations $\sigma_\alpha(t)$ for all cloud types α in the different subfigures. The SDs are shown over time for all the different models proposed in the thesis.



(g)

(h)

(i)

(j)

Figure C.3: Cloud type filling fractions standard deviations $\sigma_\alpha(t)$ for all cloud types α in the different subfigures. The SDs are shown over time for all the different models proposed in the thesis. (Cont.)

D

Code

The code for this project consists of multiple steps. Some of the scripts need to be run locally to make figures and such, and some more computationally expensive parts can be run on the VRLab. All code can be found on `https://github.com/NerineUsman/space-time-clouds`.

D.1 Data download

- **input** Date range
- **output** On the VRLab in `labdata\nerine\space-time-clouds\raw_data` . Contains cod, cth and derived motions in different sub directories for the full disk. The output directory is accompanied with a file describing the downloaded files.

The data for 1 year (dec-jan-feb) is 192 GB.

D.2 Data cleaning

Transformation of all seperate raw data files, to one image per time, which contains all desired variables at an equidistant grid.

- **input** files from `labdata\nerine\space-time-clouds\raw_data`
- **output** in `clean_data`. Contains coordinates (x, y) in m, cloud top height h, cloud optical depth d, horizontal wind velocity (u, v) in m/s, cloud type c.

After cleaning the data only 4.6 GB remain of the 192 GB.

D.2.1 cleaning steps

- **Check** which files still have to be processed.
- **Area** selection and mask area close and over land.
- **Interpolation** interpolate CTH to get same spatial resolution as COD.

- **Day-time** selection based on quality COD.
- **Quality cloud top height** Quality flag of CTH
- **Projection** Project the lat/lon coordinates to a projection in meters (UTM).
- **Advection** Find the derived motion file which is corresponding to the time. Combine the data from all different bands and apply the projection also to these data points. Use NN to interpolate the derived motions (they occur on a irregular grid) and add to data points.
- **Classification ISCCP** Use the ISCCP values to classify the pixels into different cloud types, based on CTH/COD or cloud mask from COD product.

D.3 Train data

For the models it is useful to have table with transition observations instead of the images. Per day we start from pixels which are 15 km apart and follow the cloud based on the derived motions. From this the transition pairs $s_i(t), s_i(t + \Delta t)$ are determined.

For the spatially dependent models different training sets are made in a similar way but with $s_i(t), g(s_{\{i\}}(t)), s_i(t + \Delta t)$, depending on the function g.

D.4 Models

Each of the models train some parameters based on the observed transitions pairs. These are saved in the folder `mod`. Model 1 refers to the CSS MC, Model 2 to the CSS SCA, Model 3 to the SCA and the parameters for model 0 can be found in the folder for model 1.

D.5 Simulations

Since the simulations are computationally low cost and the number of parameters defining each of the models are limited, these can be done locally. The simulation results are saved in the folder `mod`. These can then be used to make plots.

D.6 Analysis

Using the file `src/analyze_sim_results.py`. The measures from Section 3.2 *Cloud organization metrics* are calculated.

Glossary

List of Acronyms

COD	Cloud optical depth
CTH	Cloud top height
DMW	Derived Motion Winds
MC	Markov Chain
CSS MC	Conditional state space MC
SCA	Stochastic Cellular Automaton
CSS SCA	Conditional state space SCA
LES	Large Eddy Simulation

List of Symbols

$S_i(t)$	cloud state at location i and time t
$D_i(t)$	log of COD at location i and time t
$H_i(t)$	CTH at location i and time t
$Z_i(t)$	1 if there is clear sky at location i and time t and 0 o.w.
Ω	cloud state space
N	number of cloud types
$\{i\}$	set of locations belonging to the neighbourhood of location i
$p_\gamma(s,k)$	transition probability or transition density from state s to k given that the state of the neighbourhood is γ
$f^{(c)}(k\|s)$	cloud transition density function
ρ_α	cloud type α filling fraction
σ_α	standard deviation of cloud type α filling fraction over the macroscopic blocks
I_{org}	Organization Index