Improving weather and climate predictions by Cross Pollination in Time

Author:
Francine Schevenhoven
Utrecht University
3697355

Supervisors:
Dr. ir. Frank Selten (KNMI)
Prof. dr. ir. Jason Frank (UU)

June 30, 2016

Abstract

Historically, weather and climate forecasting has always been important. But the changing climate with the increasing frequency of extreme weather events and larger societal impacts has made good forecasting skill of even more interest[6]. Improving individual weather and climate models is a difficult task, but models have improved steadily over time as witnessed by objective skill scores. Recently it has been proposed to combine imperfect models dynamically in order to further improve predictions. In this thesis we explore a technique called Cross Pollination in Time (CPT, Smith 2001). In the CPT approach the models exchange states during the prediction. The number of possible predictions grows quickly with time and a strategy to retain only a small number of predictions, called pruning, needs to be developed. In the training phase a pruning strategy is proposed based on retaining those solutions that remain closest to the truth. From the training phase probabilities are derived that determine weights to be applied to the imperfect models in the forecast phase. The CPT technique is explored using low-order dynamical systems and applied to a global atmospheric model. The results indicate that the CPT approach improves the forecast quality over the individual models. The technique is suited for application to state-of-the art high-dimensional weather and climate models.

Contents

Chapter 1

Introduction

In the last decades, climate change has received more and more attention. Especially the impact of humans on climate has attracted a great deal of concern. To provide a scientific basis to develop climate-related policies, in 1988 the Intergovernmental Panel on Climate Change (IPCC) was established. The IPCC periodically publishes reports written by scientists. In their assessments it becomes clear that although the climate simulations of the climate models that are developed by various institutes in the world improve over time, these models remain imperfect. For instance, even in the simulation of the multi-decadal annual mean precipitation rate models are off by various millimeters per day in large parts of the world. Figure 1.1a [5] displays the multi-model mean of the annual mean precipitation (mm/day) for the period 1980-2005. The relative errors (Figure 1.1d) are larger than 30% in most regions, especially model values for the tropics are far off historical observations. Even more challenging is the simulation of the response of the climate to scenario's of future emissions of greenhouse gasses. The warming under the RCP4.5 scenario (a standard scenario) for 2081-2100 as simulated by 42 climate models shows a large spread (Figure 1.2). Obviously there is no consensus between the models.

Improving the models is a large research subject. A demanding aspect is that there are many unknown parameters and approximations in a model because not all physical processes are explicitly resolved. To model and tune all of these aspects requires a huge computational force effort. Models will remain imperfect due to the complexity of the climate system with interaction across scales over many orders of magnitude ($10^8 - 10^{-6}$ m). Often it helps to average across models as model errors tend to average out. In the IPCC example of the temperature in the future, the output of the models is hence averaged, a multi-model mean is calculated. In [2] this multi-model ensemble method is supported, there is some evidence it improves imperfect individual predictions. However, it is not obvious which imperfect models and with what weights the forecast should be made. Because of that almost all operational predictions are based on equal weights.

In this thesis it is proposed to combine models dynamically to further improve predictions. A training method is developed to accomplish this. This training method provides information about the strengths of each model. It is the aim to show that using these strengths can result in better predictions than the predictions of the imperfect models individually. The method is developed using chaotic low-order dynamical systems with only three degrees of freedom, that have often been used as toy models for the atmosphere. These models are computationally very cheap allowing for extensive experimentation. Secondly, the method developed on these toy models is applied to a more chaotic and realistic global atmospheric model with 1449 degrees of freedom.

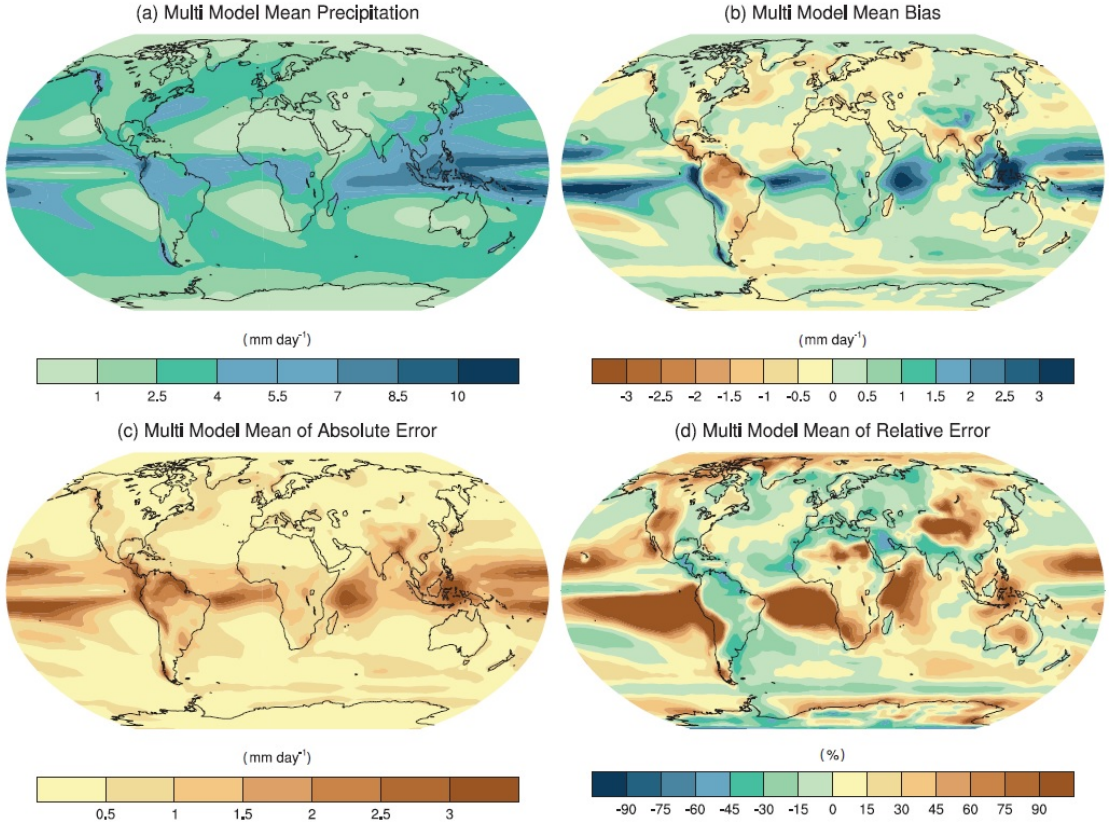


Figure 1.1: Annual-mean precipitation rate (mm/day) for the period 1980-2005. (a) Multi-model-mean. (b) Difference between multi-model mean and precipitation analyses from the Global Precipitation Climatology Project (Adler et al., 2003). (c) Multimodel- mean absolute error with respect to observations. (d) Multi-model-mean error relative to the multi-model-mean precipitation itself.

1.1 Supermodeling approach

In [13] it is proposed to combine models into one large supermodel by prescribing connections between model equations. These connections act as nudging terms to bring the solutions of the different models together. If the connection coefficients are large enough, the system will quickly synchronize on an (almost) joint trajectory. The nature of this trajectory depends on the relative strength of these connections coefficients. For instance, if all models are nudged towards one model and this one model is not nudged to the others, the ensemble will synchronize on the solution of this model. By training the connection coefficients such that the synchronized trajectory follows an observed trajectory as close as possible a supermodel is constructed with improved prediction capability. The training of the coefficients is achieved by minimization of a cost function that measures the distance between the synchronized trajectory and the observed trajectory. In [15] it is noted that the size of these coefficients is typically very large. In that case the synchronized solution of the connected ensemble of models can be reproduced by a weighted superposition of the imperfect models. In this thesis it is proposed to dynamically combine imperfect models by a new procedure to construct such a weighted superposition of imperfect models. The weights are learned from observed trajectories. This learning process follows the Cross Pollination in Time (CPT) method, proposed by Leonard A. Smith[12]. CPT "crosses" different model trajectories in order to create a larger ensemble. Hence the weighted superposition forms a supermodel that can explore larger regions of state space compared to the individual imperfect models and hence might be trained to more closely follow the truth. The method is developed using low-order dynamical systems. The model with standard parameter values generates observations. Imperfect models are created by perturbing parameter values.

1.2 Outline

Section 2 of this paper explains the concept of a dynamical system and gives a few examples that are used in this thesis, the Lorenz 63 system and the Lorenz 84 system. The next chapter explains the training by cross pollination, how the weights should be determined and how the imperfect models should be chosen. In section 4 the method is applied to the Lorenz 84 system. Section 5 shows the results for the Lorenz 63 system. CPT is also tested on a more complicated atmosphere model, this is done in section 6. In section 7 different combinations of imperfect models are tested. The last sections contain a discussion and conclusion.

Figure 1.2: Surface air temperature change in 2081-2100 for a collection of individual models, displayed as anomalies with respect to the period 1986-2005 (Figure 12.9 from the Fifth Assessment Report 2013)

Chapter 2

Dynamical systems

The supermodeling approach is tested in this thesis on low-order dynamical systems. A continuous-time dynamical system consists of a space X and a family of maps $F^t : X \to X, t \in T$, where $F^{t+s} = F^t \circ F^s$ and $F^0 = Id$[7]. A common way to define a continuous-time dynamical system is by use of differential equations. For example with $X = \mathbb{R}^3$:

$$\dot{x} = f(x), x \in \mathbb{R}^3$$

with $f : \mathbb{R}^3 \to \mathbb{R}^3$ and $\mathbb{R}^3$ forward invariant under the flow F^t of f.
The considered dynamical systems in this thesis are chaotic for certain parameter values. This means that small differences in the initial state lead to rapid divergence of orbits. For a discrete map, as used in numerical computations, formally there is sensitive dependence on initial conditions if there is an $\epsilon > 0$ such that for every $x \in X$ and $\delta > 0$ there are $y \in X$ and $n \in \mathbb{N}$ for which $d(x, y) < \delta$ and $d(f^n(x), f^n(y)) > \epsilon$[3]. Weather models are also chaotic which makes weather predictions so difficult.
Despite this chaotic behavior, the trajectories are confined to in the forward invariant set X, and the intersection of all such sets has a complex structure: a strange attractor.

2.1 Lorenz 63

In 1963 Edward Lorenz developed a simple dynamical system to model atmospheric convection[8]. The differential equations of the system depend on system parameters σ, ρ, β. The state space is described by coordinates x, y, z.

$$\dot{x} = \sigma(y - x) \tag{2.1}$$
$$\dot{y} = x(\rho - z) - y \tag{2.2}$$
$$\dot{z} = xy - \beta z \tag{2.3}$$

The standard parameter values are $\sigma = 10, \rho = 28$ and $\beta = \frac{8}{3}$. For these parameter values a strange attractor appears, with the shape of a butterfly, as depicted in figure 2.1. Varying the parameters can cause completely different behavior of the system. For example, the system can reduce to a stable fixed point. This qualitative change is called a bifurcation. If the value of ρ is varied, one can analyze that at $\rho_H = \frac{\sigma(3+\sigma+\beta)}{\sigma-\beta-1}$ a bifurcation takes place. If ρ increases through this point a chaotic attractor arises. Each"butterfly wing" contains an unstable fixed point at its center, around which the trajectories alternately revolve in an unpredictable pattern

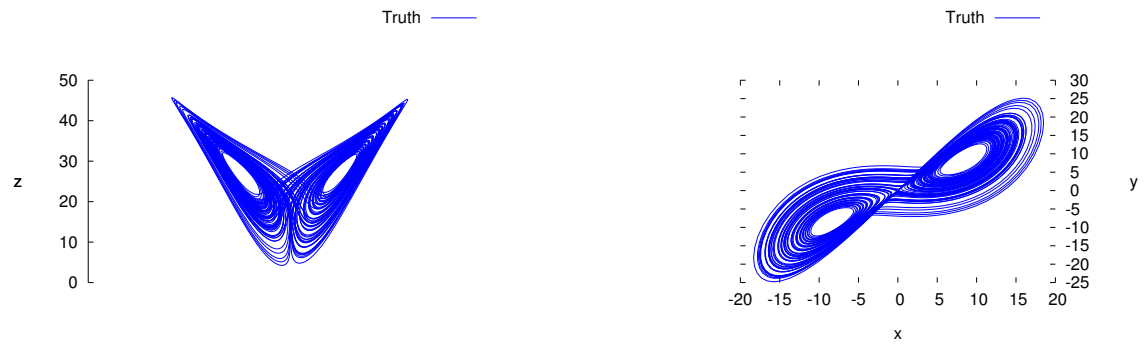


Figure 2.1: Plots of the attractor of the Lorenz 63 system with the standard parameter values.

2.2 Lorenz 84

Another dynamical system proposed by Lorenz is the Lorenz 84 system:

$$\dot{x} = -y^2 - z^2 - ax + aF \tag{2.4}$$

$$\dot{y} = xy - bxz - y + G \tag{2.5}$$

$$\dot{z} = bxy + xz - z \tag{2.6}$$

This system represents atmospheric circulation. Here x corresponds to the intensity of the zonal wind and y, z represent the cosine and sine parts of a traveling wave. The standard parameter values are $a = 0.25, F = 8.0, b = 4.0, G = 1.0$. For these values, a strange attractor appears that is more complex than the butterfly of Lorenz 63, as can be noted from figure 2.2.

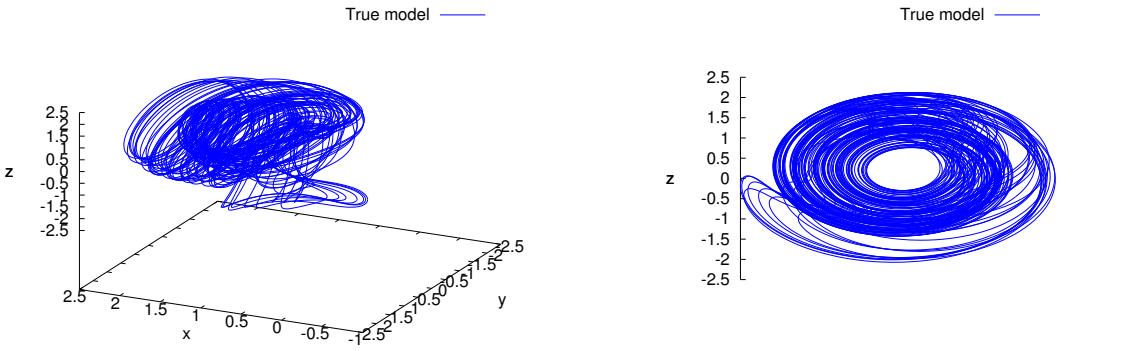


Figure 2.2: Plots of the attractor of the Lorenz 84 system with the standard parameter values.

Chapter 3

Cross Pollination in Time

Cross Pollination in Time (CPT) is a method for combining different models by "crossing" the trajectories they generate to obtain a larger ensemble. A straightforward way to use M different imperfect models is to compute M N-member ensembles. Even if the M models are comparable in quality it could be that different models perform better in different regions. To distinguish which model performs best in which region often there are not enough data available[12]. For that reason Smith presents the idea of cross pollinating trajectories between models.
It is assumed that we have an observed trajectory, called the "truth". The training phase of CPT starts from an observed initial condition in state space. From this initial state, the M imperfect models run for a certain period each ending in a different state. From these endpoints all models run again. Continuing this process leads to a rapid increase of the number of predictions with time, allowing different areas of the state space to be explored. To retain only a bounded number of predictions a pruning step is required. We choose to continue only those predictions that remain closest to the truth, the others are discarded. In figure 3.1 an example is shown for $M = 3$ different models.

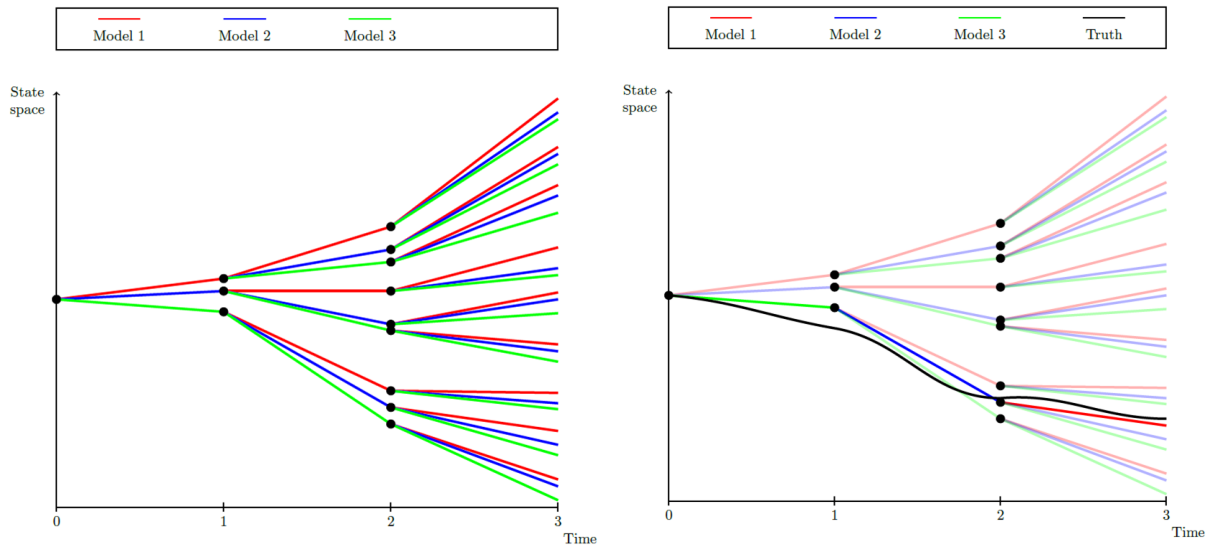


Figure 3.1: Cross Pollination in Time for 3 models, with pruning (right).

3.1 Determining weights

In the training phase, probabilities are derived to express the proportion of time for which a given model best predicts the truth for a given variable. For example, if cross pollination has taken place P times and model 1 is chosen to continue with P_1 times, the probability that model 1 remains closest to the truth is $\frac{P_1}{P}$ for P large enough. These probabilities then are used in predicting future states by integrating a weighted sum of the time-derivatives of all imperfect models with weights corresponding to these probabilities. So in the example, the weight is equal to $\frac{P_1}{P}$, hence we integrate the expected value of the time-derivative. This superposition of weighted imperfect models forms a supermodel, as all models contribute to the evolution.

3.2 Iterative method

In order to obtain convergence towards a supermodel that reflects the truth in the best possible way the training is carried out iteratively. The first iteration step is to obtain a weighted sum of the imperfect models, the supermodel. In the second iteration step, this supermodel is added as an extra imperfect model. In the subsequent iteration steps the previously obtained supermodel is replaced by the newly obtained supermodel. If the added supermodel is closer to the true model than the initial imperfect models, the constructed trajectory in the CPT procedure needs fewer strong adjustments from the initial imperfect models. The trajectory will remain closer to the observations, which can result in better estimations of the weights. Learning stops if the supermodel remains closer to the truth than the individual imperfect models for all time steps during the training.

3.3 CPT as a parameter estimation method

In both the Lorenz systems, the imperfect models in this thesis will satisfy the same differential equations as the associated perfect model. Only the parameters are varied. Because the parameters appear linearly in the equations, it is possible to reproduce the perfect model with a weighted sum of imperfect models exactly, but only if the imperfect parameter values satisfy certain conditions. These conditions are derived in this subsection.

As mentioned in section 2, the equations for the Lorenz 84 system are as follows:

$$\dot{x} = -y^2 - z^2 - ax + aF \tag{3.1}$$
$$\dot{y} = xy - bxz - y + G \tag{3.2}$$
$$\dot{z} = bxy + xz - z \tag{3.3}$$

In the equation for $\dot{x}$, both an a and $a \cdot F$ appear. Therefore, if should there only be two imperfect models for the equation for $\dot{x}$ for example: $\dot{x}_1$ with parameters a_1 and F_1, and $\dot{x}_2$ with parameters a_2 and F_2, it would not always be possible to get a weighted average of $\dot{x}_1$ and $\dot{x}_2$ in order to produce $\dot{x}$. In this case there are three constraints whilst there are only two equations for $\dot{x}$. In a matrix-vector equation this can be represented as follows, where β_1 and β_2 denote the weight given to $\dot{x}_1$ and $\dot{x}_2$:

$$\begin{pmatrix} a_1 & a_2 \\ a_1F_1 & a_2F_2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} a \\ aF \\ 1 \end{pmatrix}$$

This system will only give a solution for the standard parameter values for a and F if the equations for a and aF are linearly dependent and if a_1, a_2 are not both equal to zero. Since a and F are not known, it is unlikely that the parameters will be chosen this way. Hence we need to add an extra linearly independent equation for $\dot{x}$ with a_3 and F_3:

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ a_1F_1 & a_2F_2 & a_3F_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} a \\ aF \\ 1 \end{pmatrix}$$

Three equations, three unknowns, this should do the job in case not two or more elements of $\{a_1, a_2, a_3\}$ are zero. Or at least, it will give real values for β_1, β_2 and β_3. But since the CPT weights are positive by definition, another constraint applies: $\beta_1, \beta_2, \beta_3 \geq 0$. The following definitions are introduced:[1]:

Definition. *Let $\boldsymbol{x}^1, ..., \boldsymbol{x}^k$ be vectors in $\mathbb{R}^n$ and let $\lambda_1, ..., \lambda_k$ be nonnegative scalars whose sum is unity.*
(a) *The vector $\sum_{i=1}^{k} \lambda_i \boldsymbol{x}^i$ is said to be a **convex combination** of the vectors $\boldsymbol{x}^1, ..., \boldsymbol{x}^k$.*
(b) *The **convex hull** of the vectors $\boldsymbol{x}^1, ..., \boldsymbol{x}^k$ is the set of all convex combinations of these vectors.*

This convex hull generalizes the 'in between' concept for one dimension: if we want a linear combination of ξ_1 and ξ_2 with positive weights to reproduce parameter ξ, ξ has to lie in between ξ_1 and ξ_2. To be able to reproduce the n-dimensional vector $\mathbf{x}$, it has to lie inside the convex hull of vectors $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^k$. Hence we know that for n parameters that appear linearly in one differential equation for a state variable, $n + 1$ linearly independent vectors of these parameters are needed which form a convex hull around the true parameter vector. A convex hull constructed in this way is called a simplex.

Chapter 4

Results for Lorenz 84

The first CPT experiment is performed with the Lorenz 84 system. As observed truth a trajectory with the standard parameter values is used. From section 3.3 about the convex hulls it is clear that one needs at least three imperfect models to possibly get a linear combination of the models that represents the truth. As can be seen in table 4.1, the parameter values of the three imperfect models we used differ substantially from the truth. Figure 4.2 shows the very distinct attractors of the imperfect models. Figure 4.1 shows that a produced CPT trajectory remains close to the observations for a training period of $T = 1000$ time steps, where in every time step a pruning step is taken and only one solution is continued. Numerical solutions are obtained by using a fourth-order Runge Kutta time stepping scheme, with a time step of 0.01.

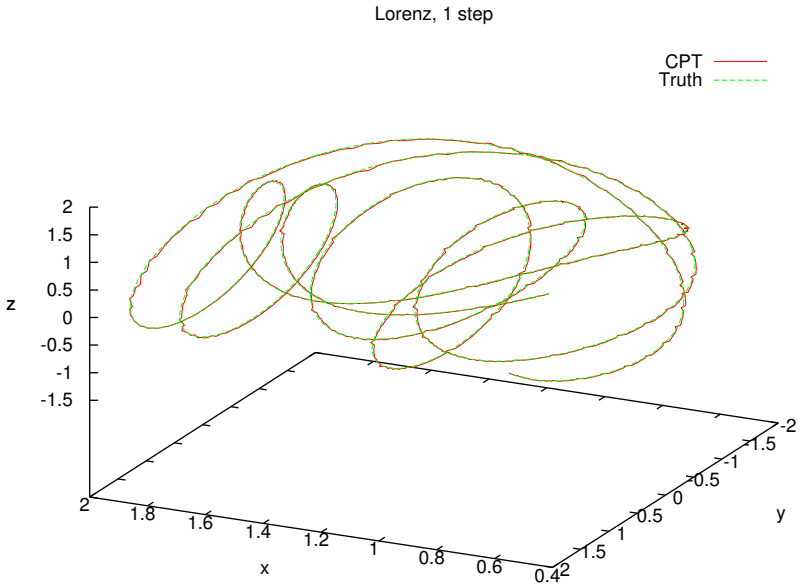


Figure 4.1: CPT training period with $T = 1000$ for the Lorenz 84 system.

For this experiment the models are integrated for $T = 100$ time steps, for 100 iterations. In every iteration the same part of the attractor is used.

Model	a	F	aF	b	G
Model 1	0.1	4.8	0.48	3.0	0.4
Model 2	0.3	4.8	1.44	3.0	1.2
Model 3	0.3	15.0	4.50	5.5	1.2
True	0.250	8.00	2.00	4.00	1.00

Table 4.1: Standard and perturbed parameters values for the Lorenz 84 system.

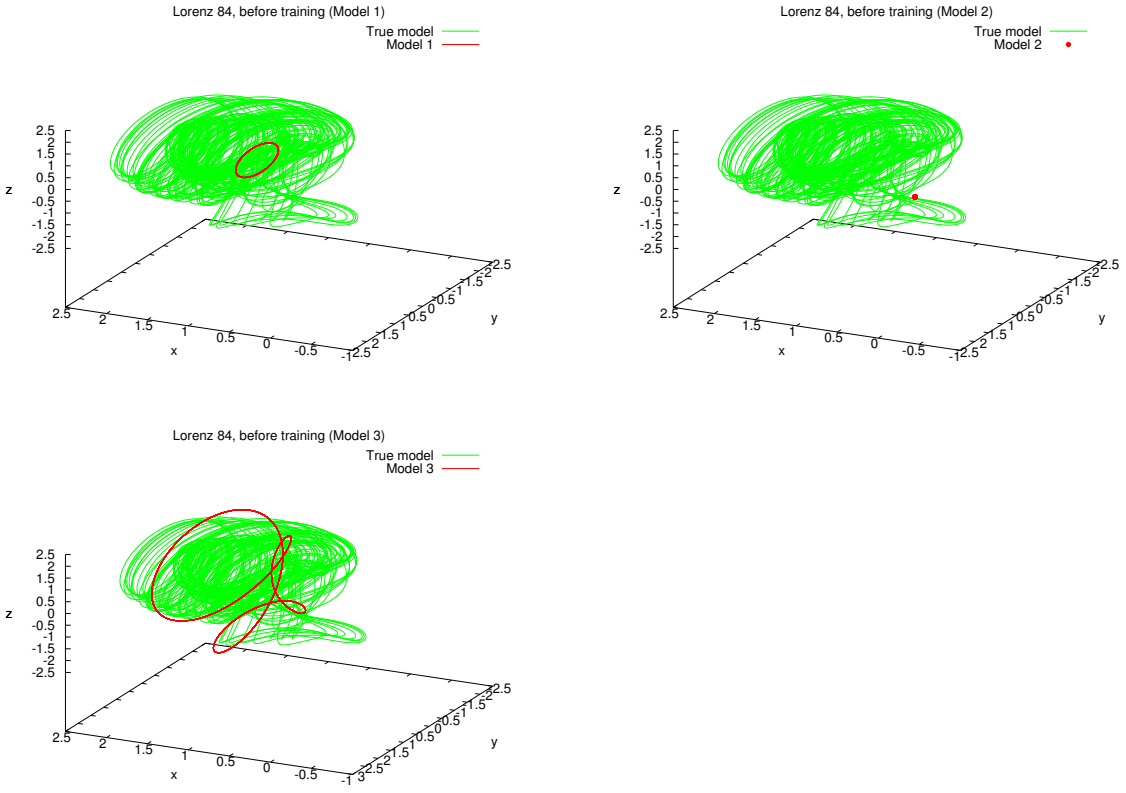


Figure 4.2: Trajectories of the imperfect models (red), together with the true trajectory (green) of Lorenz 84.

Already after one iteration, the weights for a, aF are almost perfect and remain the same during the other 99 iterations. For b, G the iteration steps are quite helpful. After one iteration the weight for model 1 is too small in the y component, the weight factor should be about twice as large. But after 100 iterations the weight factor resembles quite well the perfect weight factor and the weights also have stabilized up to a few decimals. The weights w_i, $i \in \{1, 3\}$ after the CPT procedure are presented in table 4.2. They determine the superposition of the imperfect models.

$$\dot{x}_{super} = \sum_{i=1}^{3} w_i^x \dot{x}_i \tag{4.1}$$

$$\dot{y}_{super} = \sum_{i=1}^{3} w_i^y \dot{y}_i \tag{4.2}$$

$$\dot{z}_{super} = \sum_{i=1}^{3} w_i^z \dot{z}_i \tag{4.3}$$

Model	w_i^x	w_i^y	w_i^z
$i = 1$	0.2100	0.2506	0.5068
$i = 2$	0.5400	0.4904	0.0000
$i = 3$	0.2500	0.2509	0.4932

Table 4.2: Weights of the supermodel of the Lorenz 84 system.

In the case of the Lorenz 84 system, the superposition of the imperfect Lorenz 84 models again forms a Lorenz 84 system, because of the linearity of the parameter values a, aF, b, G in the differential equations. Hence the supermodel is a Lorenz 84 system for which the parameter values can be calculated, where the value of b is chosen to be the average of b_y and b_z.

$$a_{super} = \sum_{i=1}^{3} w_i^x a_i \tag{4.4}$$

$$aF_{super} = \sum_{i=1}^{3} w_i^x aF_i \tag{4.5}$$

$$b_{super} = \frac{1}{2}\sum_{i=1}^{3} w_i^y b_i + \frac{1}{2}\sum_{i=1}^{3} w_i^z b_i \tag{4.6}$$

$$G_{super} = \sum_{i=1}^{3} w_i^y G_i \tag{4.7}$$

	a	F	aF	b	G
True	0.250	8.000	2.000	4.000	1.000
Supermodel	0.258	7.765	2.003	4.005	1.000

Table 4.3: Parameter values of the truth and the supermodel for the Lorenz 84 system.

If this supermodel is integrated for a long time period, the attractor of the supermodel and the truth look quite similar, as can be seen in figure 4.3.

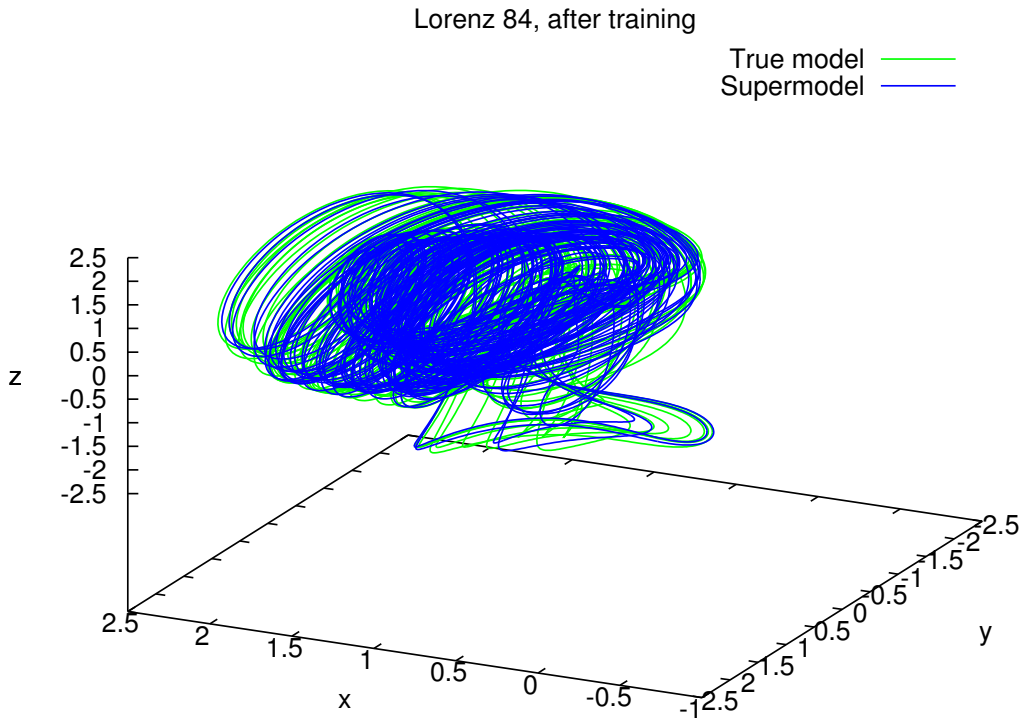


Figure 4.3: Trajectory of the supermodel (blue), together with the true trajectory (green) of the Lorenz 84 system.

4.1 Climate measures

Straightforward measures to compare the attractor of the supermodel and the truth are the mean (μ), standard deviation (σ) and covariance.

$$\mu(x) = \frac{1}{T}\sum_{t=1}^{T} x(t) \tag{4.8}$$

$$\sigma(x) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x(t) - \mu(x))^2} \tag{4.9}$$

$$\mathrm{cov}(x, y) = \frac{1}{T}\sum_{t=1}^{T}(x(t) - \mu(x))(y(t) - \mu(y)) \tag{4.10}$$

The calculation of these statistics is based on 500 runs of 5000 time steps. Also the error estimation of a 95% confidence interval is calculated. It is assumed that if there are enough runs, their mean, standard deviation and covariance will approximate a normal distribution. For this normal distribution a 95% confidence interval can be calculated: $(\bar{x} - z^* \frac{\sigma}{\sqrt{n}}, \bar{x} + z^* \frac{\sigma}{\sqrt{n}})$ Here $n = 500$, and $(\bar{x}, \sigma)$ are the parameters of the normal distribution. To have a 95% confidence interval, $z^* \approx 1.96$. From table 4.4 it is clear that the attractors of the true model and the supermodel have almost the same statistics. Only for mean z, Cov. yz and Cov. xz small but significal differences are found.

	Truth	Supermodel
Mean x	1.01831 (0.00375)	1.01581 (0.00411)
Mean y	0.05343 (0.00806)	0.06158 (0.00922)
Mean z	0.27277 (0.00256)	0.26605 (0.00290)
SD x	0.57770 (0.00671)	0.57688 (0.00781)
SD y	0.91427 (0.00119)	0.91331 (0.00117)
SD z	0.90700 (0.00104)	0.90630 (0.00098)
Cov. xy	-0.04208 (0.00814)	-0.05025 (0.00931)
Cov. xz	-0.03996 (0.00170)	-0.03494 (0.00183)
Cov. yz	-0.07120 (0.00284)	-0.08110 (0.00332)

Table 4.4: The mean, standard deviation (SD) and covariance for the Lorenz 84 system for the truth and the supermodel. Between brackets the 95% error estimation is given.

4.1.1 Kullback-Leibler

As measure for the difference between attractors the Kullback-Leibler divergence (KL)[10] can be used. This measure quantifies to what extent a distribution $P(x)$ is approximated by a distribution $Q(x)$. Here, $P(x)$ denotes the 'truth'. It is defined as follows[10]:

$$KL(P||Q) = -\int P(x) \log \frac{Q(x)}{P(x)} dx$$

Clearly, if $Q = P$, then $KL(P||Q) = 0$. Further, $KL(P||Q) \geq 0$, as can be seen with Jensen's inequality. If $Q(x) = 0$ but $P(x) > 0$ the log term blows up. Because of that a smoothing is introduced[10]: $K(x) = \alpha \exp(-\frac{\|x\|}{2h^2})$, where α is chosen such that $\int K(x)dx = 1$, hence $K(x)$ becomes a Gaussian distribution. The exact distributions $P(x)$ and $Q(x)$ are not known, instead only a sequence of values in time. For $P(x)$ this sequence is denoted by $x(t_n)_{n=1}^N$ and for $Q(x)$ this is denoted by $y(t_n)_{n=1}^N$. Then the distributions $P(x)$ and $Q(x)$ can be approximated as follows:

$$P(x) \approx \frac{\alpha}{N} \sum_{n=1}^{N} \exp(-\frac{\|x - x(t_n)\|^2}{2h^2})$$

and

$$Q(x) \approx \frac{\alpha}{N} \sum_{n=1}^{N} \exp(-\frac{\|x - y(t_n)\|^2}{2h^2})$$

Substituting these terms in the log term gives a convenient expression because α vanishes. Then only the smoothing parameter h has to be chosen. But the $P(x)$ term in front of the log term is more difficult to calculate with this expression of superposition of Gaussians because of the α term. However, since the truth $P(x)$ is a sequence of values in time, it can be represented with delta functions: $P(x) = \frac{1}{N} \sum_{n=1}^{N} \delta(x - x_n)$. The sum and integral can be interchanged because of convergence. So the KL divergence becomes: $-\frac{1}{N} \sum_{n=1}^{N} \int \delta(x - x_n) \log \frac{Q(x)}{P(x)} dx$. From the fundamental properties of the Dirac delta it is known that $\delta(x - x_n)$ only has a value for $x = x_n$, otherwise it is equal to zero. And since the function is defined as a distribution, also $\int \delta(x - x_n)dx = 1$ holds. Therefore, $-\frac{1}{N} \sum_{n=1}^{N} \int \delta(x - x_n) \log \frac{Q(x)}{P(x)} dx$ reduces to $-\frac{1}{N} \sum_{n=1}^{N} \log \frac{Q(x_n)}{P(x_n)}$. So $P(x)$ is as factor in front of the log represented by a superposition of delta functions, and in the log term approximated by the superposition of Gaussians.

Now it is possible to calculate the Kullback-Leibler distance:

$$KL(P||Q) \approx -\frac{1}{N} \sum_{n=1}^{N} \log \left(\frac{\sum_{k=1}^{N} \exp(-\frac{\|x(t_n) - y(t_k)\|^2}{2h^2})}{\sum_{k=1}^{N} \exp(-\frac{\|x(t_n) - x(t_k)\|^2}{2h^2})} \right)$$

The sequence $x(t_n)$ consists of 50000 subsequent points from the true model. The sequence of observations $y(t_n)$ consist of 50000 subsequent points from the supermodel. With some trial and error h is chosen to be equal to 0.1. If h is too small, $h = 0.01$ for example, $KL(P||Q)$ still goes to infinity. If h is too large,

for example $h = 1$, then $KL(P||Q)$ is almost equal to zero, even if the distributions of the true model and the imperfect model are quite different.

Model	$KL(P\|\|Q)$
Model 1	12.17
Model 2	55.25
Model 3	20.61
Supermodel	0.11
Truth	0.065 (0.028)

Table 4.5: Kullback Leibler value for 50000 subsequent points. Between brackets the standard deviation for $KL(P||Q)$ is given.

Table 4.5 shows some more evidence that the supermodel has an attractor that is more similar than those of the true model than the initial imperfect models. But to be able to interpret the $KL(P||Q)$ values a bootstrapping method is performed. The true model runs from 100 different initial conditions on the attractor for 50000 time steps. The run from the first initial condition is taken as truth. The $KL(P||Q)$ value is then calculated 99 times. If it is assumed this bootstrapping method gives $KL(P||Q)$ values that approximate a normal distribution, a standard deviation can be calculated. Then, the average true $KL(P||Q)$ value is less than two standard deviations smaller than the $KL(P||Q)$ for the supermodel.

4.2 Forecast quality

Along with a measure for the climate statistics of the models, also a measure for quality or skill of a "weather prediction" can be constructed. This measure reflects the forecast quality of the models on shorter time scales. The squared Euclidean distance between the true trajectory and the trajectory of a model with a slightly perturbed initial condition is calculated and averaged over a number of forecasts, as shown in figure 4.4. On the true attractor, this value saturates for large enough forecast time T at a value corresponding to the average distance between two arbitrary states. This distance is used to normalize the measure of the forecast quality. The initial perturbation is chosen on the order of 10^{-1}. The number of forecasts is equal to 10000 and $d = 10$ time steps.

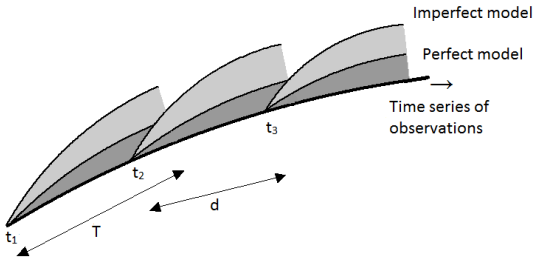
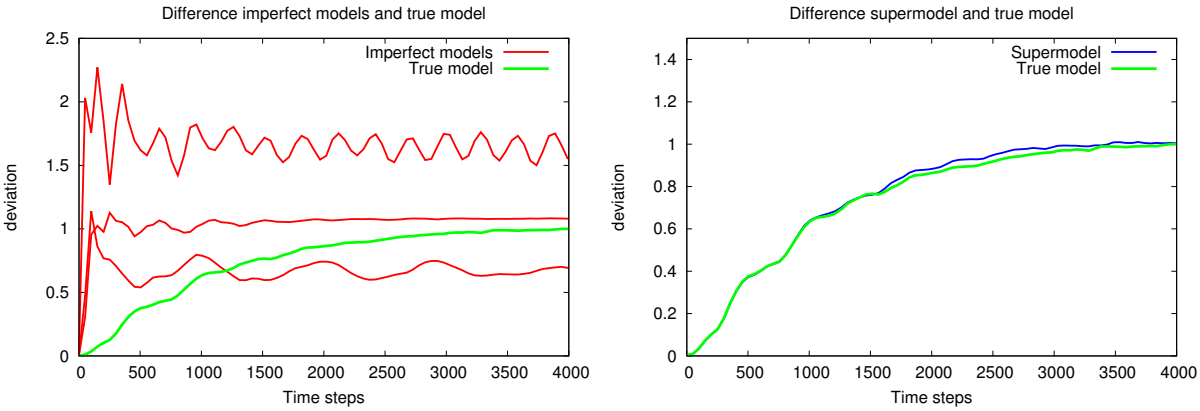


Figure 4.4: Measure of the forecast quality. At times t_i a short integration of time T starts from observed initial conditions and slightly perturbed conditions. The fixed time interval between times t_i is denoted by d.



(a) Imperfect models and true model

(b) Supermodel and true model

Figure 4.5: Forecast quality of the imperfect Lorenz 84 models and the Lorenz 84 supermodel (right).

It can be concluded from figure 4.5 that the imperfect models almost immediately have no added value for the predictability. The supermodel definitely performs better. Up to 1500 time steps, where the scaled difference is 0.8, there is no significant difference between the forecast quality of the supermodel and the true model. After 1500 time steps the supermodel performs slightly worse but its performance is still is very similar to that of the perfect model.

Chapter 5

Results for Lorenz 63

The CPT approach is next tested on another low-order dynamical system, the well-known Lorenz 63 system. Two different imperfect models are created with parameter values that deviate about 30% from the standard parameter values.

	σ	ρ	β
Truth	10	28	$\frac{8}{3}$
Model 1	12.25	19	3.3
Model 2	7.5	35	1.9

Table 5.1: Standard and perturbed parameter values for the Lorenz 63 system.

The behavior of these imperfect models is quite different from the truth, as depicted in figure 5.1. The first model has as attractor a stable fixed point. The second model has a chaotic attractor that resembles the truth although there is a strong bias, the mean z-value is considerably larger.

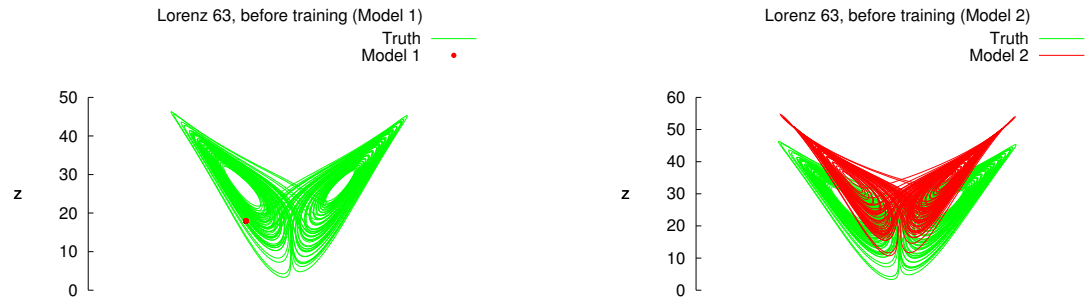


Figure 5.1: Trajectories of the imperfect models (red), together with the true trajectory (green) of the Lorenz 63 system.

In the CPT training phase in every time step it is determined which model lies closest to the truth. Only the trajectory of this model is continued. The training phase T is chosen to be 200 time steps. The number of iterations is 100. Every iteration step the same part of the attractor is used for training.
The weights w_i, $i \in \{1, 2\}$ that are the result of the training phase are listed in table 5.2. They determine

the superposition of the imperfect models.

$$\dot{x}_{super} = \sum_{i=1}^{2} w_i^x \dot{x}_i \tag{5.1}$$

$$\dot{y}_{super} = \sum_{i=1}^{2} w_i^y \dot{y}_i \tag{5.2}$$

$$\dot{z}_{super} = \sum_{i=1}^{2} w_i^z \dot{z}_i \tag{5.3}$$

Model	w_i^x	w_i^y	w_i^z
$i = 1$	0.5248	0.4385	0.5491
$i = 2$	0.4752	0.5615	0.4509

Table 5.2: Weights of the supermodel of the Lorenz 63 system.

After 45 iterations, the weights for $\dot{y}$ and $\dot{z}$ do not change anymore. The weights for $\dot{x}$ are after 100 iterations still not converged, but the values differ only from the third decimal onwards.
Also in this case, the superposition of the imperfect Lorenz 63 models again forms a Lorenz 63 system, because of the linearity of the parameter values σ, ρ, β in the differential equations. Hence the supermodel is a Lorenz 63 system for which the parameter values can be calculated:

$$\dot{\sigma}_{super} = \sum_{i=1}^{2} w_i^x \dot{\sigma}_i \tag{5.4}$$

$$\dot{\rho}_{super} = \sum_{i=1}^{2} w_i^y \dot{\rho}_i \tag{5.5}$$

$$\dot{\beta}_{super} = \sum_{i=1}^{2} w_i^z \dot{\beta}_i \tag{5.6}$$

	σ	ρ	β
Truth	10	28	$\frac{8}{3}$
Supermodel	9.993	27.983	2.669

Table 5.3: Parameter values of the truth and the supermodel of the Lorenz 63 system.

If this supermodel is integrated for a long time period, the attractor of the supermodel and the truth look quite similar, as can be seen in figure 5.2.

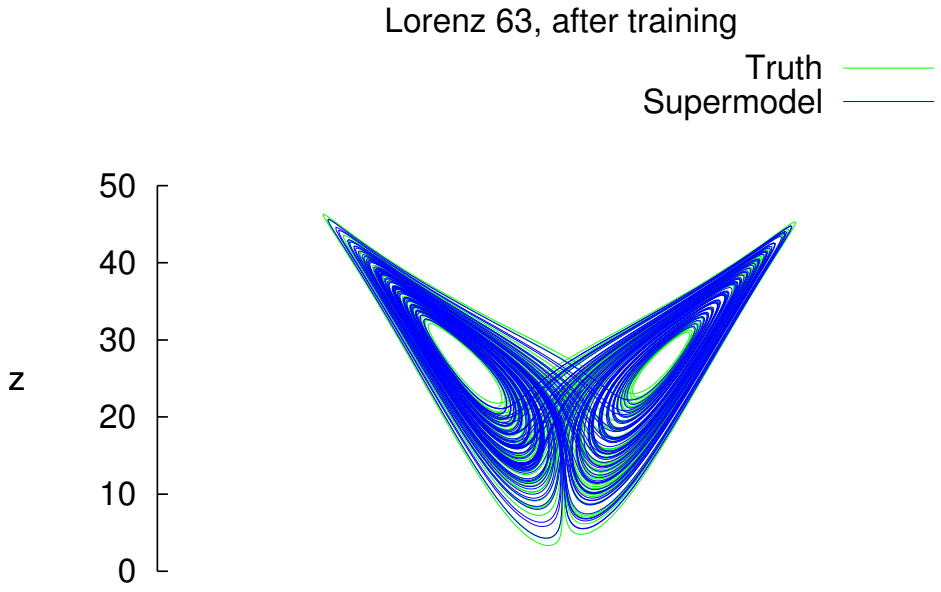


Figure 5.2: Trajectory of the supermodel (blue), together with the true trajectory (green) of the Lorenz 63 system.

5.1 Climate measures

The calculation of the statistics of the model trajectories (mean, standard deviation and covariance) is based on 500 runs of 5000 time steps. Also the error estimation of a 95% confidence interval is calculated. The statistics of both the true and the supermodel attractor are very similar (Table 5.4). Especially the standard deviations for both x, y and z do have up to the first decimal the same value. The largest differences are in the covariance between x, z and y, z. However, these differences are 95% within the uncertainty intervals and are thus not significant. The sizes of all confidence intervals for both the truth and the supermodel are almost identical.

	Truth	**Supermodel**
Mean x	0.073 (0.099)	0.033 (0.098)
Mean y	0.073 (0.099)	0.034 (0.098)
Mean z	23.552 (0.012)	23.528 (0.012)
SD x	7.843 (0.010)	7.844 (0.009)
SD y	8.939 (0.011)	8.942 (0.010)
SD z	8.618 (0.012)	8.623 (0.012)
Cov. xy	61.529 (0.150)	61.547 (0.148)
Cov. xz	0.189 (0.266)	0.057 (0.268)
Cov. yz	0.247 (0.336)	0.109 (0.334)

Table 5.4: The mean, standard deviation (SD) and covariance for the Lorenz 63 system for the truth and the supermodel. Between brackets the 95% error estimation is given.

5.1.1 Kullback-Leibler

Another measure to compare attractors of different models is the Kullback-Leibler measure. For the Lorenz 63 system this is calculated similarly as for Lorenz 84, only the value of h has to be adapted. For Lorenz 84 h was chosen to be equal to 0.1, but for Lorenz 63 this causes values of $KL(P||Q)$ going to infinity for the imperfect models. Hence h is chosen to be larger, $h = 10$. From table 5.5 it can be concluded that the supermodel cannot be distinguished from the true model with respect to the Kullback-Leibler measure.

Model	$KL(P\|\|Q)$
Model 1	0.56
Model 2	0.15
Supermodel	$1.50 \cdot 10^{-3}$
Truth	$3.62 \cdot 10^{-3}$ $(1.98 \cdot 10^{-3})$

Table 5.5: Kullback Leibler value for 50000 subsequent points. Between brackets the standard deviation for $KL(P||Q)$ is given.

5.2 Forecast quality

As was done for the Lorenz 84 system, the forecast quality of the Lorenz 63 system is measured with the mean squared error of forecasts started from slightly perturbed initial conditions. The initial perturbation is chosen on the order of 10^{-1}. The number of forecasts is equal to 1000 and $d = 10$ time steps. Figure 5.3 shows that the ability of the supermodel and the true model to predict the observed truth is about the same. The imperfect models almost immediately lose their ability to predict well.

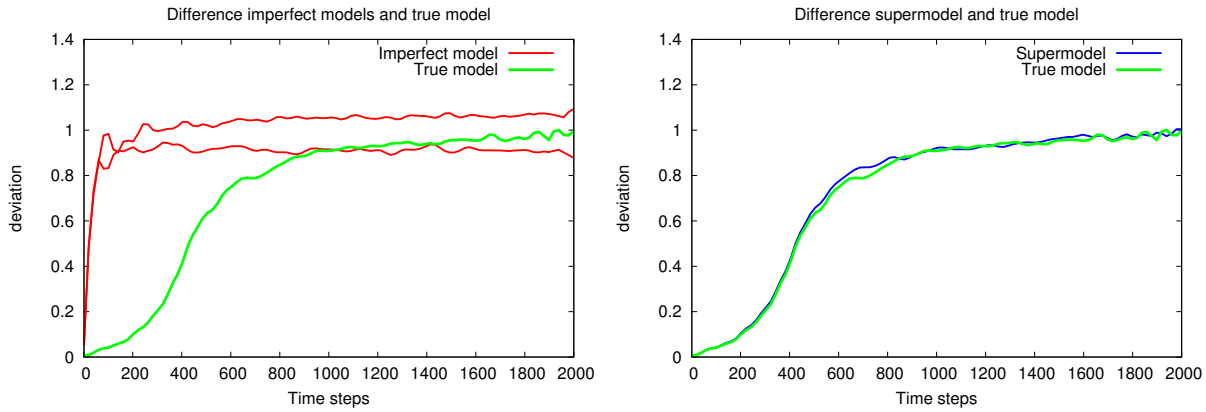


Figure 5.3: Forecast quality of the imperfect Lorenz 63 models (left) and the Lorenz 63 supermodel (right).

Chapter 6

Results for a quasi-geostrophic model

The Cross Pollination in Time method is next applied to the three-level quasi-geostrophic model of atmospheric flow on a sphere of Marshall and Molteni[9]. The three levels are 200 hPa, 500 hPa and 800 hPa. The model integrates equations for the quasi-geostrophic potential vorticity (q) at these three levels. A geostrophic model assumes that the Coriolis force is balanced by the pressure gradient force. The potential vorticity can be seen as a measure of the circulation of an air parcel. The potential vorticity of air parcels is conserved following the motion in the absence of forcing and dissipation: $\frac{dq}{dt} = 0$. This total time derivative $\frac{dq}{dt}$ can be written as: $\frac{\partial q}{\partial t} + \mathbf{u}\nabla q$, with $\mathbf{u}$ representing the velocity field. The velocity $\mathbf{u} = (u, v)$ can also be written in terms of the streamfunction ψ: $u = -\frac{\partial \psi}{\partial y}$ and $v = \frac{\partial \psi}{\partial x}$. Then the advection term $\mathbf{u}\nabla q$ can be written as Jacobian operator of q and ψ: $\mathbf{u}\nabla q = \frac{\partial \psi}{\partial x}\frac{\partial q}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial q}{\partial x} = J(\psi, q)$. If dissipation (D) and forcing (F) are added to the system, the following equation is obtained:

$$\frac{\partial q}{\partial t} + J(\psi, q) = D + F$$

Since the streamfunction ψ and potential vorticity q are related by a linear operator, the potential vorticity can be integrated in time if the dissipation and forcing are known.
The potential vorticity depends on longitude (λ) and latitude (ϕ) and can be represented with spherical harmonic functions:

$$q(t, \lambda, \phi) = \sum_{n=1}^{N} \sum_{m=-n}^{n} q_{mn}(t) P_n(\phi) e^{im\lambda}$$

Here, $P_n(\phi)$ denotes the Legendre polynomials. The model used supports the triangular T21 truncation. This means there are $3 + 5 + 7 + ... + (2 \cdot 21 + 1) = 22^2 - 1 = 483$ spherical harmonic coefficients q_{mn}. This means for all three levels there are $483 \cdot 3 = 1449$ spherical harmonic coefficients in total.

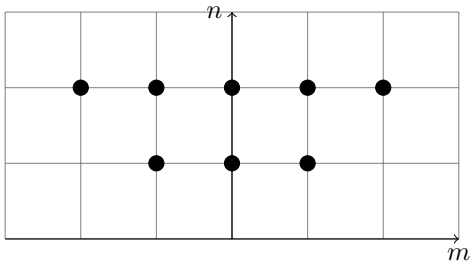


Figure 6.1: Graphic representation of a triangular truncation

6.1 Three different parameters

In the previous experiments, only parameters were varied to create different imperfect models, but the structure of the models stayed the same. In this experiment also the structure of the model is adapted, by using different forms of dissipation. To create different imperfect models, three parameter values are varied:

- $h0$ scale height of the topography in km
- *addisl* Parameter of the land-seamask dependent Ekman damping (more friction over land)
- *addish* Parameter of the orography dependent Ekman damping (more friction over steep slopes)

If the parameters $addisl, addish$ are both equal to zero, then uniform dissipation is used. Four different imperfect models are used for the CPT training phase. One of them has uniform dissipation instead of spatially varying dissipation.

	$h0$	$addisl$	$addish$
Truth	9.0	0.5	0.6
Model 1	6.0	1.0	0.0
Model 2	6.0	0.0	1.0
Model 3	14.0	0.0	0.0
Model 4	14.0	1.0	1.0

Table 6.1: Parameter values of the imperfect QG-models.

At every time step in the CPT training phase it is determined which model lies closest to the truth. The training phase T is chosen to be 100 time steps. This corresponds with an integration period of 3 days. Most of the physical processes are captured within 3 days. The number of iterations is 20. In every iteration step a new part of the attractor is used for training, to get a better sampling of the attractor. This is accomplished by continuing the integration of the truth. The time step is 1/36 day.
Since the parameters $h0, addisl$ and $addish$ only act on the lowest level in the atmosphere, there are only four weights w_i to be determined. The weights are determined for every spherical harmonic coefficient individually and then they are averaged to obtain w_i. After 20 iterations, the weights are not completely converged, they can differ a few percent per iteration.

Model	w_i
$i = 1$	0.222
$i = 2$	0.239
$i = 3$	0.243
$i = 4$	0.296

Table 6.2: Weights of the QG-supermodel.

6.1.1 Climate measures

The potential vorticity calculated by the quasi-geostrophic model can be used to determine the geostrophic winds $\vec{u}$ (m/s). Here $\vec{u} = (u, v)$, with u the zonal velocity component and v the meridional velocity component. The size of the wind strength $\|\vec{u}\| = \sqrt{u^2 + v^2}$ is used as measure for the long-term behavior of the quasi-geostrophic model. The true model, imperfect models and supermodel are integrated over 900 days in winter. In figure 6.2 and 6.3 this averaged wind strength at 800 hPa for both the true model and the supermodel is shown. As statistical measure the root mean squared error (RMSE) of the averaged wind strength over 900 days at 800 hPa is used. Since the parameters presumably will have more influence on the wind strength at land than at sea, the area of N points where the RMSE is

calculated is over land. The square consists of a part of Asia, including the Himalaya: from 25N to 60N and from 60E to 110E, as shown in figure 6.4.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\|\vec{u}_i^{truth}\| - \|\vec{u}_i^{mod}\|)^2}$$

The computed RMSE indicates that the attractor of the supermodel resembles the truth quite well, see table 6.3. To assess the uncertainty of the RMSE values again a bootstrapping technique is performed. For 98 different initial conditions a trajectory of 900 days is computed with the true model. Then the RMSE is calculated for these trajectories with respect to one other true trajectory of observations. The 95% percentile of these values is listed in table 6.3.

Model	**RMSE wind strength**
Model 1	0.54
Model 2	1.67
Model 3	4.69
Model 4	1.81
True model	0.65 (0.93)
Supermodel	0.67

Table 6.3: The root mean squared error over 900 winter days. For the true model, the average RMSE is given. Between brackets, the value is given for which 95% of the RMSE values is below that value.

Model 1 and the supermodel are indistinguishable from the true model in RMSE value, while the RMSE value of the other imperfect models is much larger.

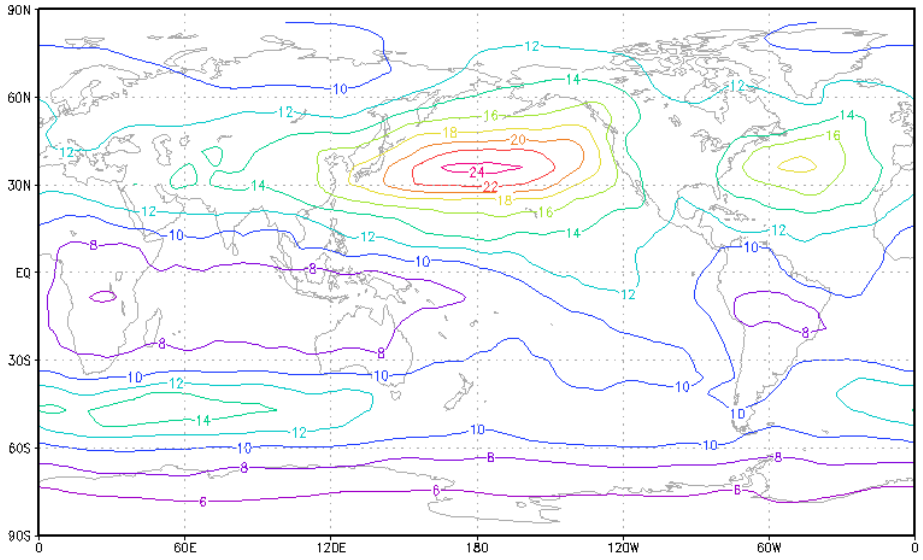


Figure 6.2: Wind strength for the true model at 800 hPa averaged over 900 days.

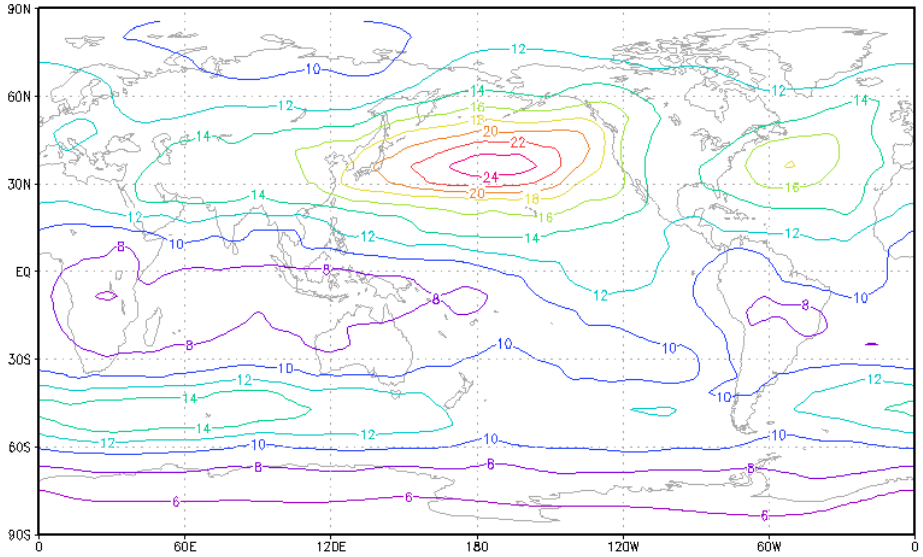


Figure 6.3: Wind strength for the supermodel at 800 hPa averaged over 900 days.

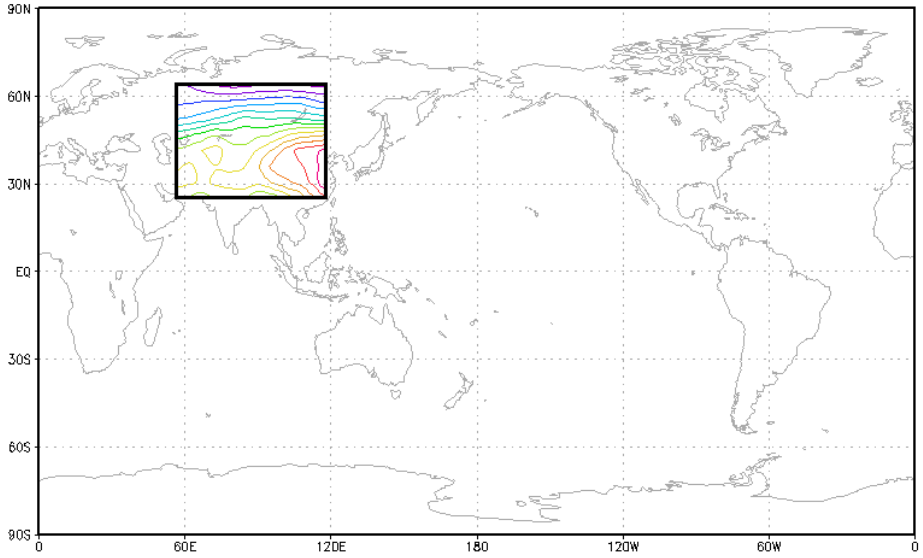
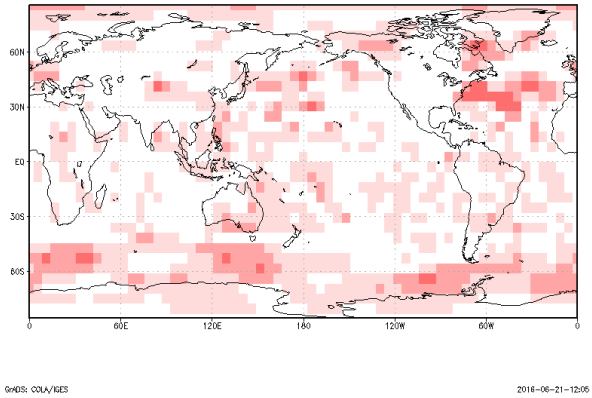


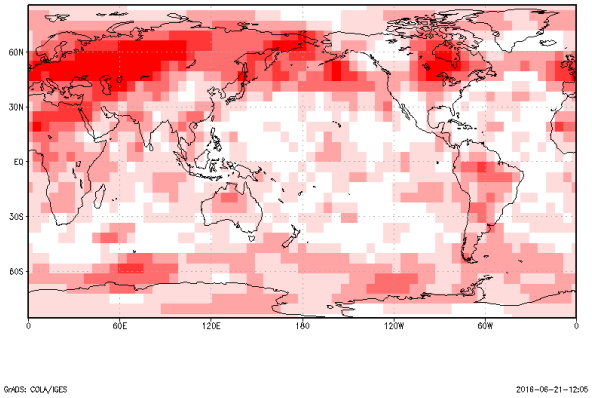
Figure 6.4: The square denotes the area for which the RMSE of the wind strength is calculated.

To prove that the impact of the parameters over land is larger than over sea, at every grid point in the world the RMSE value is calculated and compared to the mean RMSE value of the true model. In figure 6.5 can be seen that especially model 3 with the uniform dissipation has RMSE values over land

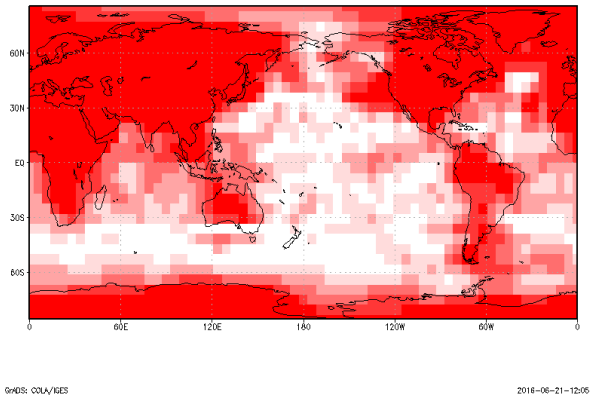
more than three times as high as the mean RMSE for the true model.



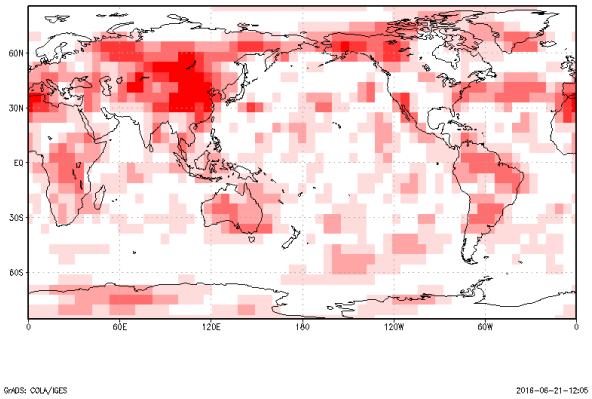
(a) Model 1



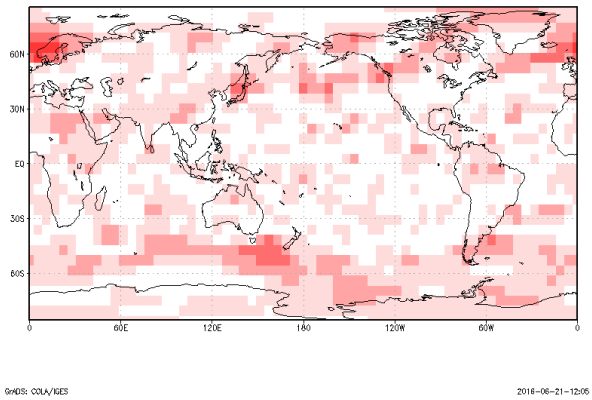
(b) Model 2



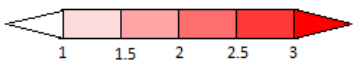
(c) Model 3



(d) Model 4



(e) Supermodel



(f) Colorbar: the color indicates how many times larger the model RMSE is compared to the mean RMSE for the true model.

Figure 6.5: Comparing the model RMSE with the mean RMSE for the true model for every grid point in the world.

6.1.2 Forecast quality

As was done for the Lorenz systems, the forecast quality is measured by calculating the mean squared error between the true trajectory and the trajectory of a model with a slightly perturbed initial condition and to average this error over a number of forecasts. The mean squared error is taken over all three levels and all spectral coefficients. The number of forecasts is 100 and $d = 1000$ time steps. The initial perturbation is chosen on the order of 10^{-4}.

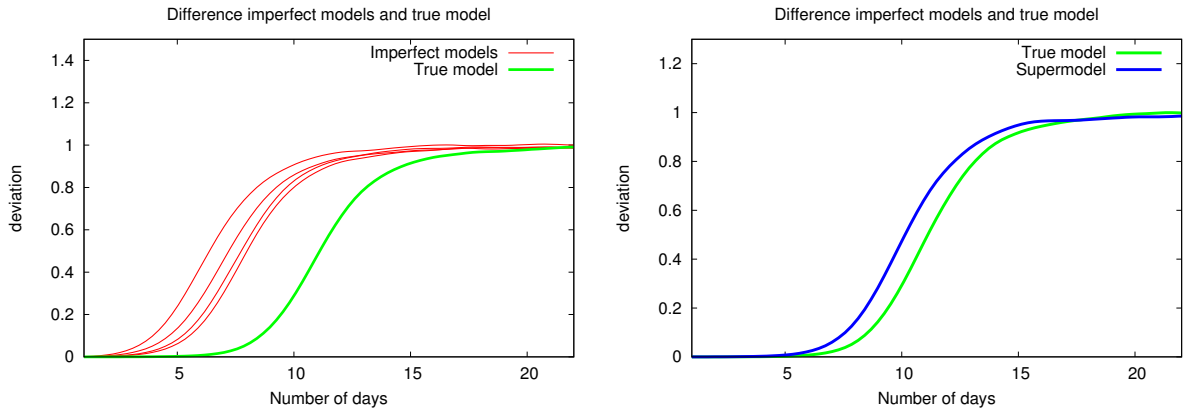


Figure 6.6: Forecast quality of the imperfect QG-models (left) and the QG-supermodel (right).

From figure 6.6 it is clear that the forecast quality of the supermodel has improved a lot compared to the forecast quality of the imperfect models.

6.2 Variation in more sensitive parameters

In the previous section, the parameter values for $h0, addisl$ and $addish$ were varied. From the imperfect models alone, it could be noticed that the value of these parameters does not have to be very precise to make a good model. For example, the imperfect values of $addisl$ and $addish$ were chosen twice as large as the true values and still the models had reasonable forecast quality. In this section other parameters are tested, for which it is known that their true value is of more importance, solutions are more sensitive to perturbation in these parameters.

- $tdis$ Time scale in days of the Ekman damping
- $rrdef1$ Rossby radius of deformation of the 200-500 hPa layer
- $rrdef2$ Rossby radius of deformation of the 500-800 hPa layer

Ekman damping is the damping on vorticity in the boundary layer[4]. The Rossby radius of deformation is the length scale at which the rotation of the earth becomes important. In table 6.4 the parameter values for the imperfect models are denoted. For the Rossby radii the deviation is only a few percent.

	$tdis$	$rrdef1$	$rrdef2$
Truth	2.0	0.1150	0.0720
Model 1	1.5	0.1165	0.0705
Model 2	1.5	0.1130	0.0725
Model 3	2.4	0.1130	0.0705
Model 4	2.4	0.1165	0.0725

Table 6.4: Parameter values of the imperfect QG-models.

The CPT procedure is carried out the same as for the previous experiment with $h0, addisl, addish$. After 20 iterations, the weights (Table 6.5) are not completely converged, they differ a few percent per iteration, but there is no increasing or decreasing trend.

Model	w_i^{200}	w_i^{500}	w_i^{800}
$i = 1$	0.653	0.217	0.093
$i = 2$	0.347	0.459	0.235
$i = 3$	0.000	0.157	0.215
$i = 4$	0.000	0.167	0.457

Table 6.5: Weights of imperfect QG-models at 200, 500 and 800 hPa.

6.2.1 Climate measures

Again as statistical measure the root mean squared error (RMSE) is used over the averaged wind strength $\|\vec{u}\| = \sqrt{u^2 + v^2}$ over 900 days in winter, at 800 hPa, but now at each of the N two-dimensional grid points on the sphere:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\|\vec{u}_i^{truth}\| - \|\vec{u}_i^{mod}\|)^2}$$

Model	**RMSE wind strength**
Model 1	1.92
Model 2	1.80
Model 3	1.10
Model 4	1.42
True model	0.48 (0.65)
Supermodel	0.45

Table 6.6: The root mean squared error over 900 winter days. For the true model, the average RMSE is given over 98 different trajectories. Between brackets, the value is given for which 95% of the RMSE values is below that value.

Table 6.6 reveals that the RMSE of the supermodel is even smaller than the average RMSE of the true model, hence the supermodel is with respect to this climate measure indistinguishable from the true model. The RMSE value of the imperfect models is much larger, also much larger than the upper bound of the 95% interval.

6.2.2 Forecast quality

To have a measure for the forecast quality, the mean squared error is again taken over all three levels and all spectral coefficients. The number of forecasts is 100 and $d = 1000$ time steps. The initial perturbation is chosen in the order of 10^{-4}. The forecast quality in figure 6.7 of the supermodel is not as good as the forecast quality of the true model but the supermodel improves the predictability significantly as compared to the individual models.

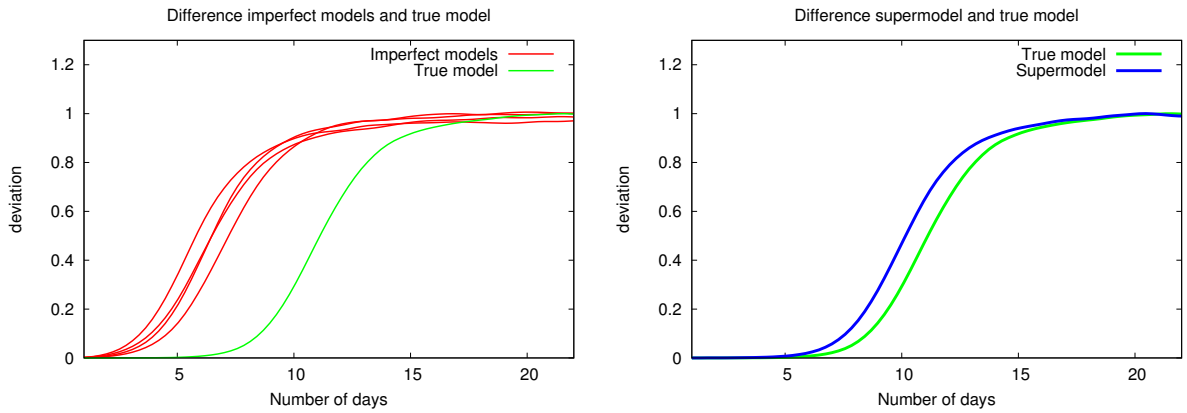


Figure 6.7: Forecast quality of the imperfect QG-models (left) and the QG-supermodel (right).

6.2.3 Use of fewer imperfect models

The imperfect model with the worst forecast quality will be left out in the next experiment. The same imperfect models from the previous subsection are used. The model with the worst forecast quality is model 1, hence the supermodel is constructed out of models 2,3 and 4. The remaining models together still contain the same different imperfect parameter values for $tdis, rrdef1$ and $rrdef2$ as before. The CPT training phase is again performed with $T = 100$ and 20 iterations.
The forecast quality in figure 6.8 of the new supermodel has improved compared to the forecast quality of the imperfect models, but clearly the forecast quality of the supermodel constructed of all four imperfect models is better. This indicates that also an imperfect model with poorer forecast quality than the other imperfect models can still contribute towards a superior supermodel.

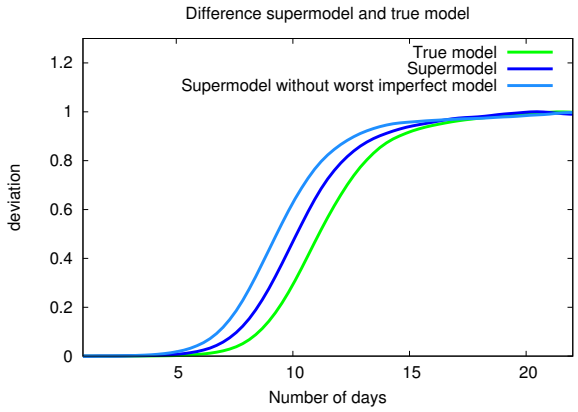


Figure 6.8: Forecast quality of the QG-supermodel and the QG-supermodel generated without the worst imperfect model in forecast quality.

6.2.4 Use of equal weights

The CPT method gives us output weights that determine a superposition of models. But to what extent are the values of these weights important? If the models all get equal weight, does that affect the quality of the supermodel? In figure 6.9 the forecast quality of a supermodel with equal weights is plotted. The equal weight supermodel does perform better than the imperfect models, but performs obviously worse than the supermodel with the weights obtained from CPT. Hence, the CPT method training adds value to the quality of the supermodel.

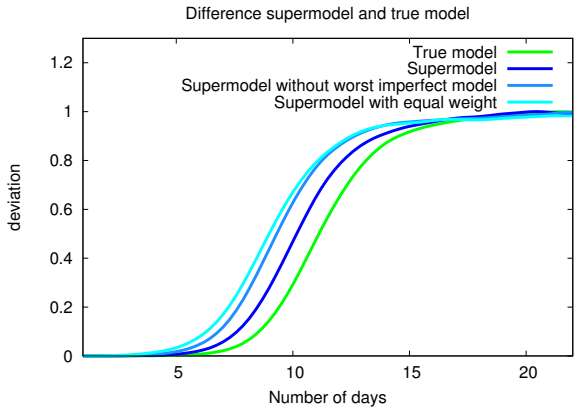


Figure 6.9: Forecast quality of the QG-supermodel, the QG-supermodel generated without the worst imperfect model in forecast quality and the QG-supermodel generated with equal weights.

Chapter 7

Different convex hulls

7.1 Lorenz 84 revisited

In the previous Lorenz 84 experiments it was assumed that the imperfect models were given such that the convex hull was already constructed. But what is the effect of using different convex hulls, all capable of forming a linear combination that represents the true model? In the next experiments we assume that for each parameter an imperfect value is given. Also we know the maximal deviation of the given values to the perfect values. For example, it is given that parameters a_1 and aF_1 deviate no more than 20% from the true values a and aF. Only it is not known whether a_1 and aF_1 are smaller or larger than a and aF. If it is assumed that a_1 is at least 80% of a, then a is at most $\frac{a_1}{0.8}$. And if a_1 is at most 120% of a, then a is at least $\frac{a_1}{1.2}$. So boundaries can be constructed where the truth must lie. For a and aF we can construct a two dimensional rectangle with the four points that are obtained this way. But since there are only two parameters, it was already known not more than three imperfect models are needed. A possibility is to include the rectangle in a triangle. This can be done in several ways, but we would like to have triangles that are as small as possible such that chosen parameter values do not deviate too much from the possible parameter values. Several triangles are constructed, like a star around the rectangle. See figure 7.1.

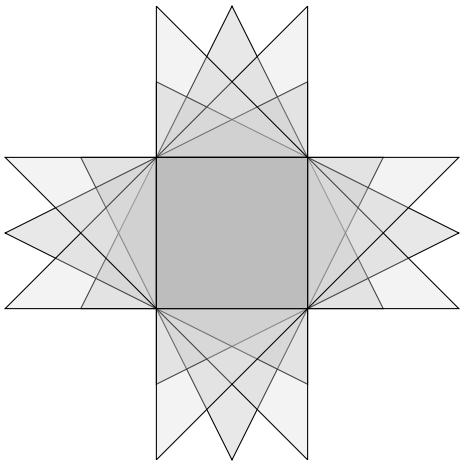


Figure 7.1: The star of eight convex hulls, where the dark grey rectangle denotes the area where the truth lies.

Now we will perform the method with the eight different convex hulls as described above. It is assumed that the parameter values for a, F, b and G are given and that they differ at most 20% from the truth. Because there are eight different convex hulls for a, aF and also eight for b, G, there are in total 64 different convex hulls possible per model. The training period is 100 time steps and the number of iterations is 50. The initial parameter values deviate less than 20% from the truth, to create convex hulls

that do not coincidentally have points that represent the truth:

	a	F	aF	b	G
Truth	0.25	8.0	2.0	4.0	1.0
Initial values	0.21	6.6	1.386	4.8	0.8

Table 7.1: Standard parameter values and perturbed parameter values for the Lorenz 84 system.

Because of the linearity of the parameters a, aF, G, b in the differential equations, the supermodel is a Lorenz 84 system for which the parameter values can be calculated, the same way as done before. With these super-parameters Lorenz 84 is integrated some time steps. The smallest difference between one of the 64 integrations and the truth decides which convex hull apparently works best on this part of the attractor. This process can be repeated for different parts of the attractor. If all convex hulls would give approximately equivalent results for $a_{super}, b_{super}, F_{super}, G_{super}$, then the probability distribution of which convex hull performs best on some part of the attractor should be uniform. The integration period with $a_{super}, b_{super}, F_{super}, G_{super}$ is 100 time steps. This is done 1000 times, where for every new integration the initial starting point on the true attractor is shifted 15 time steps forward. A histogram plot in figure 7.2 shows the non-uniformity of the convex hulls that perform best.

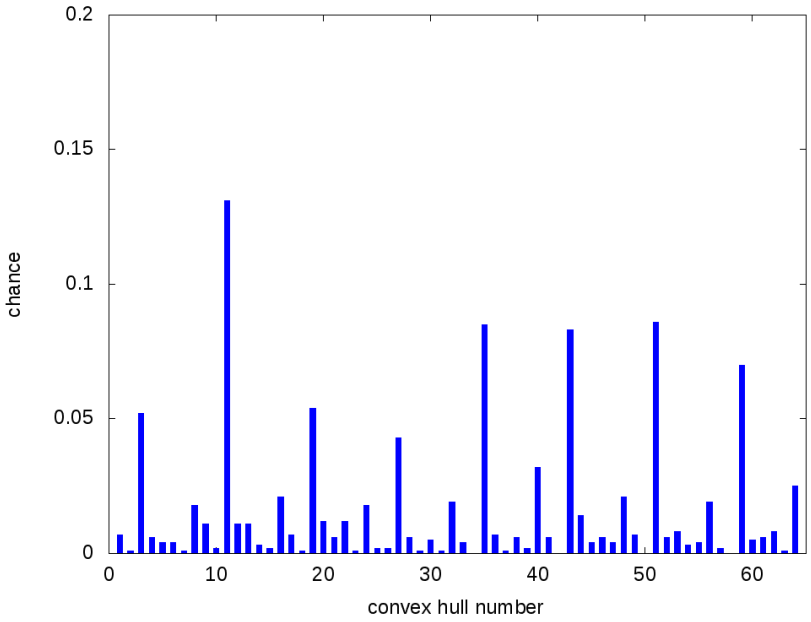


Figure 7.2: Histogram of the best performing convex hulls.

Apparently it makes much difference which convex hull is used. Why are some convex hulls performing better than others? If a convex hull has a point that lies close to the truth, then it seems reasonable that it performs better. However, in the histogram in figure 7.2 convex hull number 11 has the largest weight. Convex hull 11 has for b, G a point that lies closely to $b = 4.00$ and $G = 1.00$, but for a, aF it does not have a point close to $a = 0.25$ and $aF = 2.00$:

	a	aF	b	G
Truth	0.25	2.0	4.0	1.0
Model 1	0.087	0.48	3.91	1.03
Model 2	0.263	1.45	3.91	0.34
Model 3	0.263	2.92	7.83	1.03

Table 7.2: Imperfect parameter values for convex hull 11.

So the question why some convex hulls perform much better than others remains. Although it should be mentioned that all convex hulls give an improvement compared to the initial given parameter values.

In the next experiment each of the 64 combinations of convex hulls is tested individually to investigate whether each convex hull is able to represent the truth with an iterative training period despite their different results after one iteration. The training period is again 100 time steps and there are 50 iterations. In every iteration step the attractor is shifted 30 time steps forward, to get a better sampling of the attractor.

For every convex hull the values for a and aF remain (almost) the same after one iteration and are to three decimals identical to the true values. For b and G this is different. Some convex hulls reach the true values and stay there. Other ones reach the true values almost, but then deviate again. They show some periodic behavior. After values that deviate sometimes more from the truth than the initial parameter values, the next iterations they move again into the direction of the true values and then again deviate. An explanation could be that the precise values of the parameters a and aF are of more importance than the values of b and G in order to have a good model. Once the almost perfect values for a and aF are reached, these are maintained. For b and G the explanation seems to hold that the convex hulls that do converge the truth lies close to one of the three points of the convex hull. The other two points therefore have a larger deviation from the truth than for other convex hulls. Because of that, the other two points will not be chosen too quickly. See figure 7.3, where the black dot represents the true values. The left triangle will deviate quicker from the true values than the right triangle.
This deviation for b and G from the true values can be prevented by shifting the attractor fewer steps forward every new iteration. For example, only 5 steps instead of 30. This results in convergence to the true values and once these are obtained, there will be no deviation afterwards. Shifting the attractor no steps at all for a new iteration does result in converged values, but not always the almost perfect values.

We can conclude that every individual convex hull is able to represent the truth and is also able to maintain these parameter values. Only there is some difference in convergence speed for the different convex hulls and their consistency in keeping the same parameter values. This can be explained because of the different shapes of the convex hulls. But by shifting the attractor more or fewer steps every iteration step, all convex hulls chosen in this experiment can give almost perfect supermodels.

Figure 7.3: Different convex hulls for parameters b, G, where the dot represents the true value.

7.2 Different tetrahedrons for the QG-model

Just as done in the Lorenz 84 experiment, it is assumed that some parameter values are given and that it is known to what extent the parameter maximally deviates. Then boundaries can be constructed for the parameters, so a convex hull can be formed. For this new experiment the parameters for the different time scales are varied:

- $tdis$ Time scale in days of the Ekman damping
- $trel$ Time scale in days of the temperature cooling
- $tdif$ Time scale in days of the scale selective damping at the three levels for the smallest wavenumber

Since the parameters $tdis^{-1}, trel^{-1}, tdif^{-1}$ are linearly implemented in the QG-model, it is also possible to get from the obtained weights expressions for the supermodel parameters. In case of three parameters, the boundaries form a 3-dimensional box with eight points. But since only four imperfect models are needed, a tetrahedron can be constructed around this box. A disadvantage of this method is that the deviation of the imperfect models becomes quite large compared to the truth. If the imperfect models are too far away from the truth, it will be difficult to find suitable weights. Hence the box is divided into 4-sided pyramids: tetrahedrons. See figure 7.4.

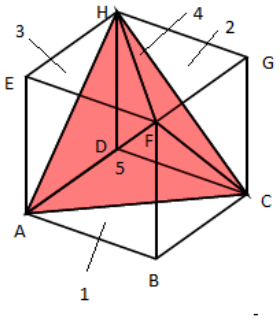


Figure 7.4: Cube with tetrahedron inside, where tetrahedron number 4 is the red one and tetrahedron number 5 the one behind the red tetrahedron.

The red coloured tetrahedron inside the box can be chosen in two ways. Once one point of the box is fixed as a point of the tetrahedron, the other three are known. This red coloured tetrahedron immediately divides the cube into five tetrahedrons.

	$tdis$	$trel$	$tdif$
Truth	2.0	40.0	5.0
Initial	1.6	32.0	4.0
Point A	1.23	24.61	3.08
Point B	2.29	24.6	3.08
Point C	2.29	45.7	3.08
Point D	1.23	45.7	3.08
Point E	1.23	24.6	5.71
Point F	2.29	24.6	5.71
Point G	2.29	45.7	5.71
Point H	1.23	45.7	5.71

Table 7.3: Parameter values of the imperfect QG-models.

The initial parameter values deviate 20% from the true values and the convex hull is constructed by assuming that the imperfect parameters deviate no more than 30%. In table 7.1 the parameter values for the truth and the eight points of the box are mentioned. The point in the box that represents the true parameter values lies closest towards precisely one of the eight boxpoints, G. The red tetrahedron inside is chosen in such a way that it does not contain this closest boxpoint. Hence there is clearly one of the five tetrahedrons that contains the truth. Although it is known that only one tetrahedron can contain the truth, all tetrahedrons are tested to investigate if they still tend to the right values. The training phase consists of 100 time steps and there are 20 iterations. The results are in table 7.2. The tetrahedron that contains the truth is indicated with a *.

Tetrahedron number	$tdis$	$trel$	$tdif$
1	2.03	30.80	4.01
2*	2.02	35.95	4.69
3	1.93	31.52	5.00
4	1.96	40.93	4.13
5	2.00	34.31	4.52

Table 7.4: Supermodel parameters for the five different tetrahedrons.

The convex hull that contains the truth, number 2, has values that approximate the true values. For the other ones, especially the value of $tdis$ has improved.

In the same way as for the other imperfect QG-models, the forecast quality is computed. The forecast quality of the imperfect models seems to be divided into two groups: A, D, E, G and B, C, F, H, where the last group has a better forecast quality. The first group has as parameter value $tdis = 1.23$ and the second group has $tdis = 2.29$, which is closer to the true value 2.0.

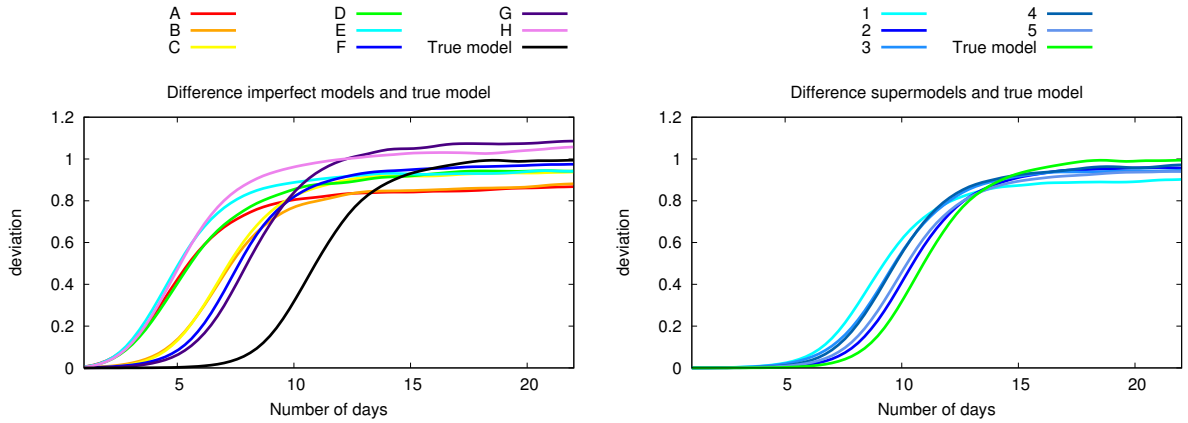


Figure 7.5: Forecast quality of the imperfect QG-models (left) and the QG-supermodels (right).

The forecast quality of the five different supermodels proves that the convex hull that contains the truth performs best, although all other supermodels give an improvement compared to the individual models A to H, while their values for $trel$ and $tdif$ still substantially differ from the true parameter values. It seems that the relative inaccuracy of parameter $tdis$ is of more importance than the inaccuracy of $trel$ and $tdif$.

7.2.1 Values for the individual spherical harmonic coefficients.

To investigate whether the different spherical harmonic coefficients react differently to the values of the parameters, the result of a CPT procedure without iteration is plotted for the spherical harmonic coefficients individually. Since it takes more time for some parameters to adapt to the system, the training period is 9000 time steps instead of the 100 time steps before. The experiment is done with tetrahedron 2, the one that contains the truth.
Immediately it can be noticed that $tdif$ has better results for higher spherical harmonic coefficients, while $trel$ has better results for lower spherical harmonic coefficients. Of course one would like to know whether the too low values for shorter integration periods have a lot of influence on the solution of the quasi-geostrophic model. Hence a sensitivity analysis is applied to the parameters $tdis, trel$ and $tdif$.

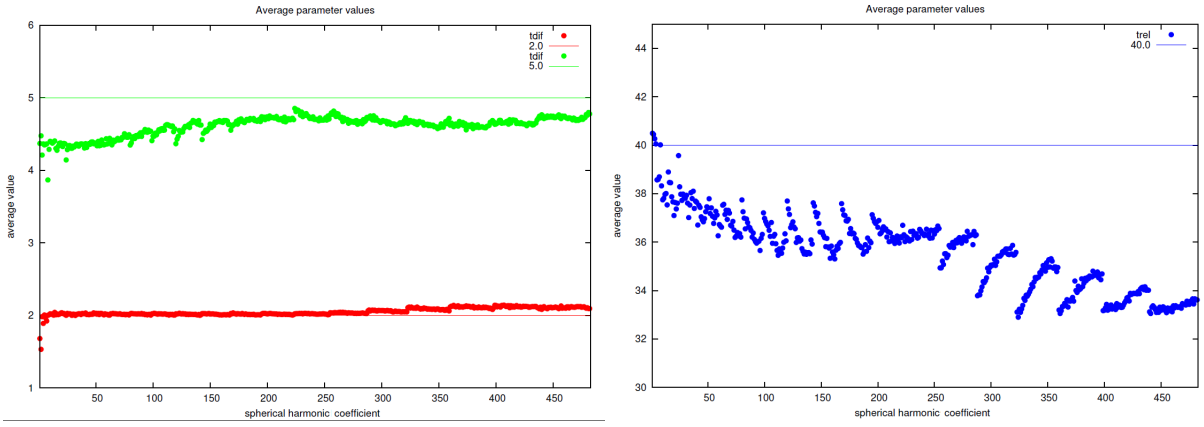


Figure 7.6: Parameter results for integration period of 9000 time steps.

7.2.2 Sensitivity analysis

The need to have exactly the right parameter values is clearly not the same for the different parameters $tdis, trel$ and $tdif$. Changing one of the input parameter values has distinct effects on the change of the output of the model. This parameter based sensitivity can be measured. Intuitively, the sensitivity S is the partial derivative of the output function with respect to the parameter[14]:

$$S = \frac{\frac{M(e_1,...,e_i+\Delta e_i,...,e_p)-M(e_1,...,e_i,...,e_p)}{M(e_1,...,e_i,...,e_p)}}{\frac{\Delta e_i}{e_i}}$$

Here M is the model output, e_i refers to the different model parameters and Δe_i is the perturbation in a single model parameter. However, this is a local method, called OAT (One At a Time). Only one parameter can be changed. But one would like to have a global, three-dimensional view of the sensitivity of $tdis, trel$ and $tdif$. Because of that in [14] the LH-OAT method is proposed, Latin Hypercube - One At a Time. The idea of a Latin Hypercube is quite simple. Every parameter represents a dimension. Each parameter is tested for a certain interval, this defines the cube. The intervals are divided into subintervals, in our case of equal length. For example, the three parameters $tdis, trel$ and $tdif$ define a three dimensional cube. We choose to have 4 subintervals. Then the Latin Hypercube consists of $4^3 = 64$ small cubes. To get a sample from this 64 cubes, in every column in every dimension exactly one cube is chosen. In total this gives 4 chosen cubes. The number of possibilities to draw such a sample is equal to $4!^{(3-1)} = 576$. If firstly the cubes are chosen in two dimensions, then for the first row there are 4 possibilities. For the second row, only 3 possibilities are left because in one column already a cube has been chosen. For the third row there are 2 possibilities, and the last one is then fixed. So in two dimensions there are $4 \cdot 3 \cdot 2 = 4! = 24$ possibilities. Then the cubes have to get placed in the third dimension, one of the four vertical layers. Again, there are $4 \cdot 3 \cdot 2 = 4! = 24$ possibilities to do that. Combined, this gives $4!^{(3-1)} = 576$ possibilities for a Latin Hypercube sample. See figure 7.7 for an example of a two-dimensional Latin Hypercube sample.
With this sampling of four hypercube points, we can for every point j for every parameter e_i perform the OAT analysis and calculate the partial effect $S_{i,j}$:[14]

$$S_{i,j} = \left| \frac{100 * \left(\frac{M(e_1,...,e_i(1+f_i),...,e_p)-M(e_1,...,e_i,...,e_p)}{[M(e_1,...,e_i(1+f_i),...,e_p)-M(e_1,...,e_i,...,e_p)]/2} \right)}{f_i} \right|$$

Here f_i is the fraction by which parameter e_i is changed.

An experiment is performed with a random sample of 4 hypercube points. As intervals for parameter ξ we choose: $[0.8\xi, 1.2\xi]$. So the subintervals have a length of 0.1ξ. If an interval is chosen for a hypercube point, the value of the parameter is the average value of the interval. The deviation fraction f_i is set to 0.05. It is randomly chosen whether the deviation of 5% is positive or negative. As model output the forecast quality is chosen as calculated in the previous section. The model with parameters $e_1, .., e_i, ..., e_p$ is regarded as true model. The Euclidean distance between the reference trajectory and the trajectory of a slightly perturbed start value is calculated, for 1000 time steps at 100 different start points. The resulting function integrated in time is M. In the matrix $S_{i,j}$, $i = 1$ stands for parameter $tdis$, $i = 2$ for parameter $trel$ and $i = 3$ for parameter $tdif$.

$$S_{i,j} = \begin{pmatrix} 168 & 79 & 88 & 113 \\ 28 & 22 & 38 & 43 \\ 28 & 27 & 17 & 33 \end{pmatrix}$$

Clearly, the sensitivity values for $tdis$ are much higher than for $trel$ and $tdif$. This is a more extended proof than in the previous section that a correct value for $tdis$ is of more importance than correct values for $trel$ and $tdif$.
Next we try a sensitivity analysis with 16 points instead of only 4. The same interval for the parameters is chosen: $[0.8\xi, 1.2\xi]$ with ξ the parameter value. The subintervals therefore have a length of 0.025ξ. The deviation fraction f_i is set to 0.01. Then the same experiment is done as for 4 points.
The results for the 16 distinct points differ a lot from each other. $tdis$ for example has sensitivity values

between 8 and 150. However, if we average the results over the 16 hypercube points, this value S_{av} is for $tdis$ clearly the highest. $S_{av}(tdis) = 57, S_{av}(trel) = 32, S_{av}(tdif) = 36$.

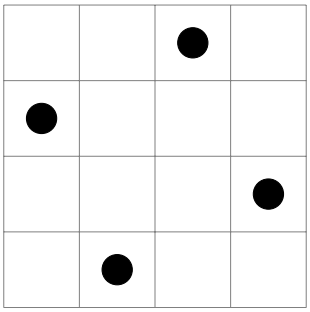


Figure 7.7: A two-dimensional Latin Hypercube sample

Chapter 8

Discussion

In this thesis the Cross Pollination in Time method is tested on low-order dynamical systems. For these models the method gave good solutions for a supermodel. However, the state-of-the-art models are far more complex than the examples from this thesis. Is the developed method also applicable to these models?

One aspect of larger models is that more computation time is needed to run the model. For the low-order dynamical systems such as Lorenz 63 and Lorenz 84, the computation time of CPT was negligible. In only a few time steps a nice approach of the true model was obtained. Also for the quasi-geostrophic model time was not an issue. Even if only a part of the attractor was used for training that was not a true reflection of the complete attractor, the method was already able to give a supermodel of good quality. There is some evidence [11] that improving weather forecasts results in better prediction of the climate for centuries. Another advantage if the training period is small, is that also the amount of observational data does not have to be large.
If the number of parameters increases, more imperfect models are needed to construct a supermodel. This will increase the required amount of computation time. Nevertheless, the increase of required imperfect models for an increasing number of parameters was found in this thesis to be linear. If every time step the number of trajectories is pruned to one single prediction, the computational cost of CPT is proportional to the number of imperfect models.

In section 2 the concept of a convex hull was introduced. The true model should lie in between the imperfect models. With only difference in parameter values for the different imperfect models this convex hull is easy to generate. For imperfect models with more differences in structure, it could become more difficult. In reality imperfect state-of-the-art weather models differ structurally and a "convex hull" approach does not apply explicity: the truth might be outside "the convex hull of the models". Adding an imperfect model that has the "opposite" behavior compared to the other imperfect models could be a solution. This added model itself does not necessarily need a quality similar to the other imperfect models. For the quasi-geostrophic atmosphere model it was concluded that also the model with the worst forecast quality contributed towards a better supermodel, if it further completes the needed convex hull.

State-of-the-art models can structurally differ in for example gridding. To perform the CPT method, regridding could be used to compare at which point in state space which model performs best. Or a common state space of the imperfect models could be defined. In this space the CPT method could be performed. With the help of data assimilation the next iteration step for the complete state space of the models could be taken.

Chapter 9

Conclusion

In this thesis a new method of combining different imperfect models dynamically was developed. This Cross Pollination in Time method was successful in creating a supermodel that in the context of simple chaotic systems resembles perfect model experiments.

To compare the quality of the supermodel with the imperfect models and the truth several measures were used. For all these measures, both "weather" measures for the short-term predictability as well as "climate" measures for the long-term predictability indicated that the obtained supermodels outperformed the imperfect individual models.

In case a weighted mean model is again a model of the same class, CPT is a parameter estimation method. In that case, the applied convex hulls for the parameters worked well. In the last section it was shown that different convex hulls are able to give good supermodels, hence the choice of the convex hulls does not have to be that precise. One requirement only is that the imperfect models surround the truth. But even if this is not the case, as also seen in the last section, the CPT method will try to optimize the parameter with the most impact on the model.

Because of the need of a convex hull, also imperfect models with less forecast quality than the other imperfect models can contribute towards a better supermodel. Removing the imperfect model with the worst forecast quality that contributes to the convex hull leads to supermodels that have substantially less quality, as observed for the quasi-geostrophic atmosphere model.

Finally it was shown for the quasi-geostrophic atmosphere model that the CPT method does have added value compared to giving each imperfect model the same weight and constructing in that way a supermodel. There are many options to apply the CPT method and to develop it further. It is not only applicable to weather and climate models, but also to other kind of models, for example economical or biological models. Therefore it is reassuring to know that compared to the tested imperfect individual models and a supermodel with equal weights Cross Pollination in Time does improve the weather and climate predictions.

Bibliography

[1] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[2] Michal Branicki and Andrew J. Majda. An information-theoretic framework for improving imperfect dynamical predictions via multi-model ensemble forecasts. *Journal of Nonlinear Science*, 25(3):489–538, 2015.

[3] M. Brin and G. Stuck. *Introduction to Dynamical Systems*. Cambridge University Press, 2002.

[4] B. Cushman-Roisin and J.M. Beckers. *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects*. International geophysics series. Academic Press, 2011.

[5] IPCC. *Climate Change 2013: The Physical Science Basis*. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.) Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1535 pp].

[6] IPCC. *Climate Change 2014: Mitigation of Climate Change*. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Edenhofer, O., R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, B. Kriemann, J. Savolainen, S. Schlmer, C. von Stechow, T. Zwickel and J.C. Minx (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

[7] Y. Kuznetsov. *Elements of Applied Bifurcation Theory*. Applied Mathematical Sciences. Springer New York, 1998.

[8] E.N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:130–140, 1963.

[9] John Marshall and Franco Molteni. Toward a dynamical understanding of planetary-scale flow regimes. *Journal of the Atmospheric Sciences*, 50(12):1792–1818, 1993.

[10] Naoya Fujiwara, Wim Wiegerinck, Frank M. Selten, and Jürgen Kurths. Sumo - supermodeling by combining imperfect models workpackage 1: Year 1. November 4, 2011.

[11] M. J. Rodwell and T. N. Palmer. Using numerical weather prediction to assess climate models. *Q.J.R. Meteorol. Soc.*, (133):129–146, 2007.

[12] Leonard A. Smith. *Nonlinear Dynamics and Statistics*, chapter Disentangling Uncertainty and Error: On the Predictability of Nonlinear Systems, pages 31–64. Mees, Alistair I., Birkhäuser Boston, Boston, MA, 2001.

[13] L. A. van den Berge, F. M. Selten, W. Wiegerinck, and G. S. Duane. A multi-model ensemble method that combines imperfect models through learning. *Earth System Dynamics*, 2(1):161–177, 2011.

[14] A Van Griensven, T Meixner, S Grunwald, T Bishop, M Diluzio, and R Srinivasan. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of hydrology*, 324(1):10–23, 2006.

[15] Wim Wiegerinck, Miroslav Mirchev, Willem Burgers, and Frank Selten. Consensus and synchronization in complex networks. chapter Supermodeling Dynamics and Learning Mechanisms, pages 227–255. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.