UNIVERSIDAD DEL NORTE

MASTER THESIS

Sequential Data Assimilation Methods for Atmospheric General Circulation Models

Author:
Randy S.
CONSUEGRA-ORTEGA

Supervisor:
Elias D. NIÑO-RUIZ, Ph.D.

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

in the

Department of Software Engineer
and
Computer Science

November 19, 2021

Declaration of Authorship

I, Randy S. CONSUEGRA-ORTEGA, declare that this thesis titled, "Sequential Data Assimilation Methods for Atmospheric General Circulation Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a master's degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. Except for such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSIDAD DEL NORTE

Abstract

Engineering Division
Department of Software Engineer
and
Computer Science

Master of Science

Sequential Data Assimilation Methods for Atmospheric General Circulation Models

by Randy S. CONSUEGRA-ORTEGA

The data assimilation (DA) process has gained some spotlight in recent years as computers have become more powerful, and models more complex. Even so, most natural phenomena have many correlations among variables that are very challenging to capture. In this proposal, we discuss the impact of an intermediate step in the leaping strategy used as a numerical integrator for Atmospheric General Circulation Models during the assimilation process, and its explicit update, particularly, for the Simplified Parameterizations, privitivE-Equation DYnamics model, nicknamed as SPEEDY. Using literature validated formulations of the Ensemble Kalman Filters the Local Ensemble Kalman Filter (LEnKF), Local Ensemble Transform Kalman Filter (LETKF), and the Ensemble Kalman Filter based on a Modified Cholesky Decomposition (EnKF-MC) experimental test are performed using the leaping step in the update process, and using only the forecast step, and letting the model propagate the updates. For the EnKF-MC formulation, we propose a formulation onto the observations space. As well, we present an intuitive Python package to perform sequential data assimilation on atmospheric general circulation models. We denote our package by Applied Math and Computer Science Lab - Data Assimilation AMLCS-DA. This package contains the efficient implementations of the previously mentioned formulations. The results reveal that our proposed framework can properly estimate model variables within reasonable accuracies in terms of Root-Mean-Square-Error when we update only the forecast state, even when using sparse operational observators (25%, 11%, 6%, 4%).

Acknowledgements

First, I want to thank God and His eternal source of wisdom and inspiration, for what this life project would not have been possible. To my beloved parents, whose support and trust inspired me into achieving it, in their beloved words is the source of my path. I am deeply grateful to advisor Ph.D. Elias Niño, for the trust placed in me and my abilities, for all the teaching, patience, and opportunities provided.

Contents

List of Figures

List of Tables

To my beloved family, colleagues, and friends

Chapter 1

Introduction

Climate forecast is of high relevance nowadays as it allows us to model the global and local climate dynamics and evolution in a given time window, which is fundamental in the prevention of natural disasters, as well as agronomic planning, among other areas of interest. As computers grow in computational power, model evaluation through multiple parametrization and time windows has proven useful to estimate and planning the utilization of natural resources, as wind parks. (Jones, Hamilton, and Wilson, 1997). To perform such forecasts, we can use Atmospheric Global Circulation Models (AT-GCM) which are a discretization of a system of dynamic equations that describes the atmospheric physics and dynamics (Hurrell, Deser, and Phillips, 2019; Teixeira et al., 2014; Bauer, Thorpe, and Brunet, 2015; Lorenc, 1986). These models are useful not only in forecasting but to understand the climate evolution at different resolutions and to identify patterns or cyclic phenomena through their evaluation. But even the best models have uncertainty associate with their estimations (noise in the source of the observation, incorrect initial conditions, computational representation, incomplete knowledge about the laws governing the dynamics of the system, and such) as is impossible to capture all the variability of real-life phenomena, this is, some level of error is associated to the model and the complex dynamics of the phenomena (Liu et al., 2019; Verstraete, Aghezzaf, and Desmet, 2020). To improve the quality of such forecasts, we can employ Data Assimilation techniques that allow us to manage that uncertainty and produce better estimations of the future state, given the dynamics of the system (Ott et al., 2004a; Saetrom and Omre, 2013; Kwiatkowski and Mandel, 2015; Wikle and Berliner, 2007) and observations. Since most of these models are computationally expensive, we require the use of High-performance computing to produce those analyses, in a reasonable amount of time (computationally speaking, as the model's parameters are in the range of $\mathcal{O}(10^8)$) (Kotsuki, Sato, and Miyoshi, 2020; Ruiz, Pulido, and Miyoshi, 2013) and efficient computational formulation that allows us to take advantage of the computational potential. Likewise, multiple AT-GCM models are available in the literature, and they use numerical integrators to compute the next state of the numerical system, so a right-hand side (rhs) of the equation is required to solve them. In a traditional differential equation, we can use directly the initial condition and a numerical integration to solve it, but in most of these atmospheric models, the rhs is coupled with the numerical integrator for the dynamic, so it can not be obtained directly. In these sorts of models, a leaping integrator is commonly used, which requires intermediate steps to solve the system. In operative schemes, we only have access to one step, which is commonly known as the model state step. In operative schemes, we only have access to one step, the model state step. In this proposal, we analyze and evaluate the use of the intermediate step in the data assimilation process for three formulations of the Ensemble Kalman Filter, using the AT-GCM model SPEEDY (Simplified Parameterizations, privitivE-Equation DYnamics),

which, in particular, uses a two steps time integrator and measure the accuracy of the updates.

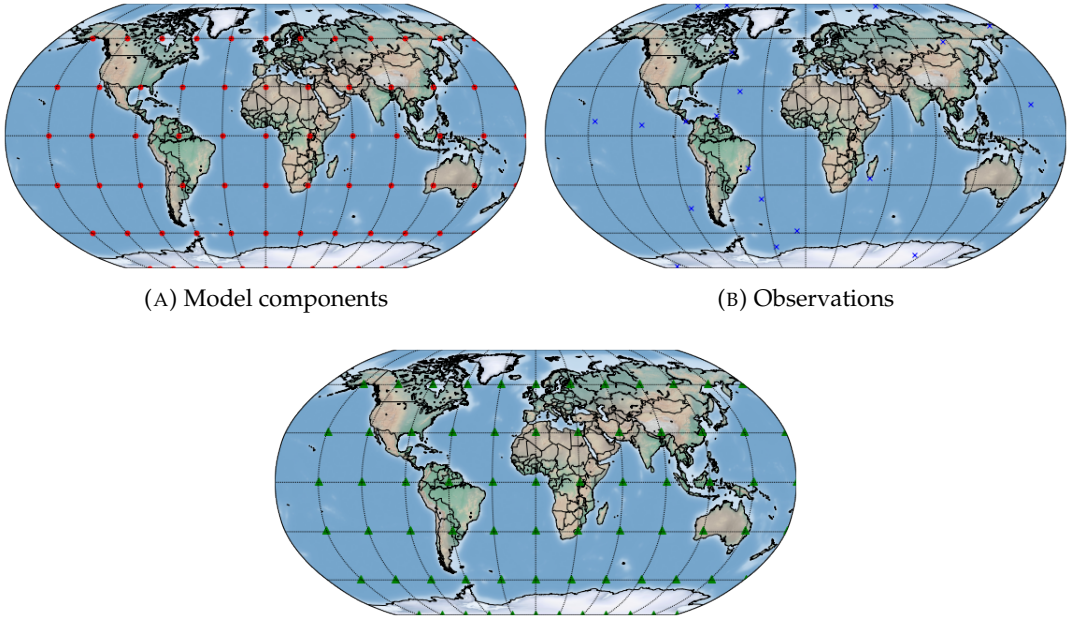
The structure of this proposal is as follows, in section 2 discusses topics related to Data Assimilation in ensemble-based methods and their formulation, as well as a brief description of the leaping integrators and the extended ensemble members to use in this proposal, in Section 3 the expected outcomes of this research and states the objective of this research are detailed, Section 4 presents our proposed method and the package developed to compare our proposal to literature validated formulations and to offer a straightforward method to use implementations with an instance of the AT-GCM model, finally Section 5 presents numerical experiments using the Atmospheric General Circulation Model SPEEDY; the results of the proposed implementation are compared against those obtained by literature formulations, as well as comparing the results using the leaping state and those that do not use it. Finally, conclusions are presented in section 6.

Chapter 2

Preliminaries

In this section, we introduce a formulation to the Data Assimilation problem, in a Bayesian framework.

2.1 The Data Assimilation Problem



(A) Model components

(B) Observations

(C) Assimilation

FIGURE 2.1: Data Assimilation Scheme

Given a model (mathematical or physical) $\mathcal{M}(\mathbf{x}_k)$ such as $\mathbf{x}_{k+1} = \mathcal{M}(\mathbf{x}_k)_{t_k \to t_{k+1}}$, where $\mathcal{M}$ is a nonlinear discretization of the real phenomena (2.1a) which can evolve dynamically in time, we can forecast the future state of the system ($\mathbf{x}_{k+1}$, also known as model realizations).

Such forecasts are affected by their chaotic and stochastic nature, so even with perfect initial conditions, the system can diverge from reality in a few steps of free run for numerous reasons, as previously mentioned.

Besides, we have some noise observations of the system state, referred to as $\mathbf{y}_k$, usually from satellites or sensors. Taking into account, not all components are observed either match with model components (as in 2.1b).

With such a model and the observations, we can obtain a better estimation of the future state by "mixing" those two sources into a new one. This process is also known as inference. Formally, we have:

$$\mathbf{x}_{k+1} = \mathcal{M}(\mathbf{x}_k)_{t_k \to t_{k+1}} + \delta \,, \tag{2.1}$$

where δ is the error of the computational model and is assumed to be unbiased ($\langle \delta \rangle = 0$). We call this model realization as *background*, which will be used as a *Prior* to estimating the future state of the system. The observations can be represented by $\mathbf{y} = \mathcal{H}(\mathbf{x}) + \boldsymbol{\epsilon_o}$ where $\boldsymbol{\epsilon_o}$ is a stochastic term (observational error) and the observation operator $\mathcal{H}(\mathbf{x}) : \mathbb{R}^{n \times 1} \to \mathbb{R}^{m \times 1}$ which maps from states space to the observation space. The observations are assimilated into the model forecast (background) by inference, using the Bayes theorem (Chen, 2003; Vetra-Carvalho et al., 2018):

$$\mathcal{P}(\mathbf{x}|\mathbf{y}) = \frac{\mathcal{P}(\mathbf{x}) \cdot \mathcal{P}(\mathbf{y}|\mathbf{x})}{\mathcal{P}(\mathbf{y})} \,, \tag{2.2}$$

Where:

- $\mathcal{P}(\mathbf{x})$ previous knowledge about the state of the system (*Prior*).
- $\mathcal{P}(\mathbf{y}|\mathbf{x}) = \mathscr{L}(\mathbf{x}|\mathbf{y})$ quantifies the distribution of errors of observations.
- $\mathcal{P}(\mathbf{y}) = \int \mathcal{P}(\mathbf{y}|\mathbf{x}) \cdot \mathcal{P}(\mathbf{x}) d\mathbf{x}$ is a normalizing constant .
- $\mathcal{P}(\mathbf{x}|\mathbf{y})$ gives the updated estimate of the true state, also called *Posterior*

Assuming that both are normally distributed. This gives us the best estimation (optimal state) $\mathbf{x}^*$, based on the observations $\mathbf{y}_k$ and the model realization $\mathbf{x}_k^b$

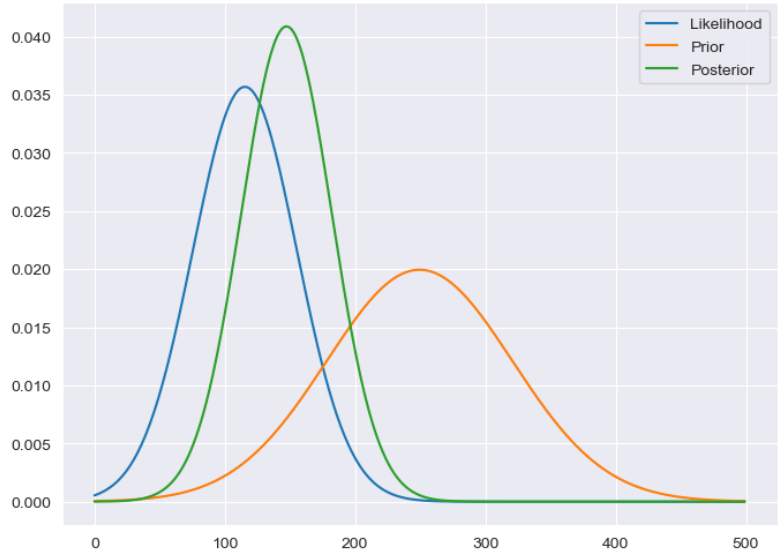
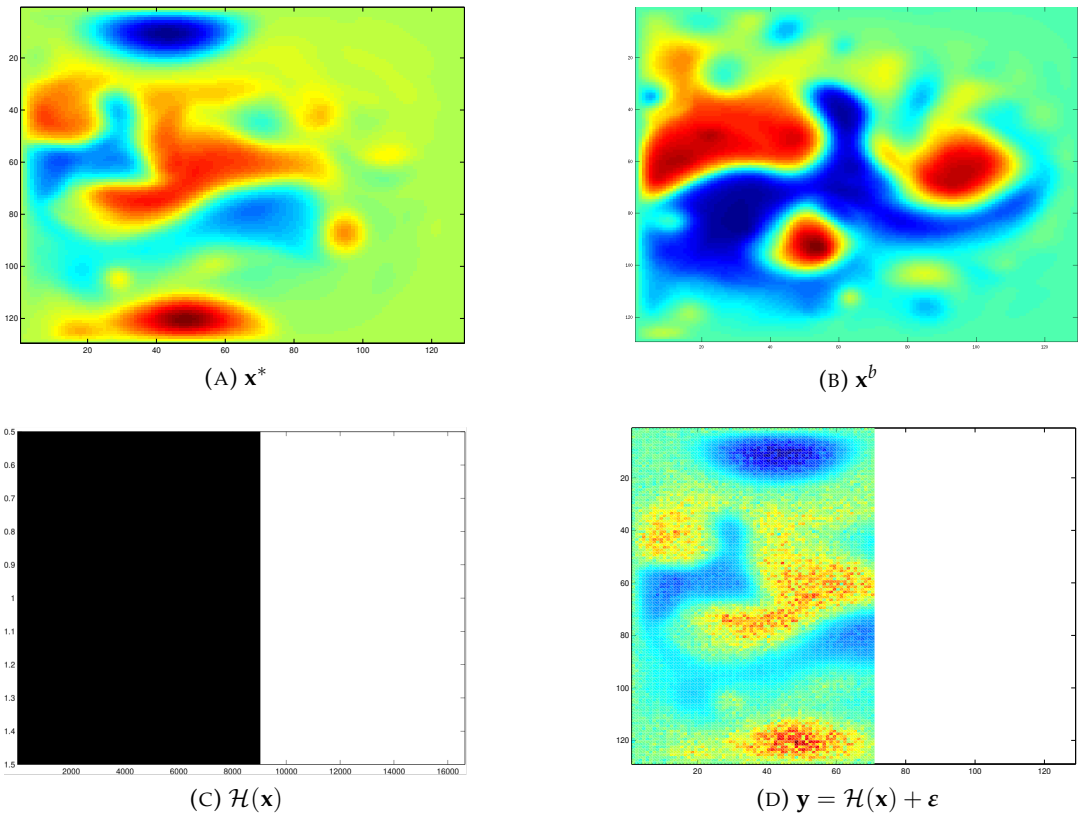


FIGURE 2.2: Bayesian inference

This is known as Data Assimilation: the process by which an imperfect numerical forecast $\mathbf{x}_k^b$ is adjusted according to real noisy observations $\mathbf{y}_k$ (Nino-Ruiz, 2018; Nino-Ruiz, Cheng, and Beltran, 2018), where $\mathbf{x}_k^b \in \mathbb{R}^{n \times 1}$ and $\mathbf{y}_k \in \mathbb{R}^{m \times 1}$ are the background state and the observations at step k, for $0 \leq k \leq M$, where, n is the model size (also known as model resolution), m denotes the number of observations per

assimilation step and M is the size of the assimilation window (the number of times wherein observations are available).



(A) $\mathbf{x}^*$ (B) $\mathbf{x}^b$

(C) $\mathcal{H}(\mathbf{x})$ (D) $\mathbf{y} = \mathcal{H}(\mathbf{x}) + \varepsilon$

FIGURE 2.3: Model forecast based on observations

By this technique, and the Gaussianity assumption, a close formula for minimizing $\mathcal{P}(\mathbf{x}|\mathbf{y})$ i.e, minimize the difference between observations and forecasts, is already known. We call this estimation *Analysis*. In this context, we solve the inverse problem, in which we use the observations to estimate the better initial condition, which minimizes the error. We can notice this combination of observations and the model as a Hidden Markov Model (HMM) as the future state only depends on the previous one.

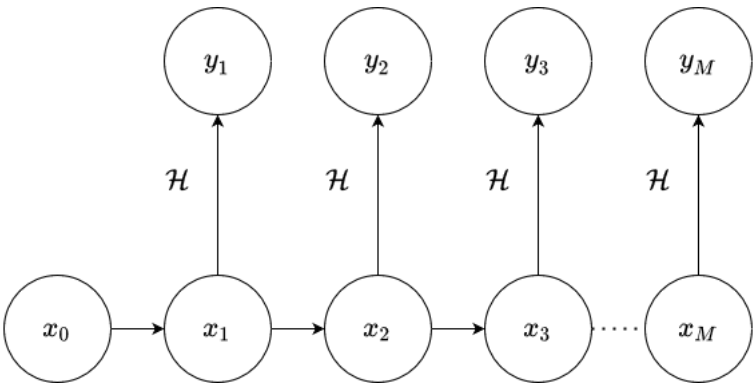


FIGURE 2.4: HMM

Recall that equation 2.2 is used to quantify the uncertainty in our model (Berger

and Smith, 2019) via model updates. When doing these updates, we expect to perform small updates into the model forecasts. If updates are too far different from the forecast, we can suspect the model is not performing as expected. We can perform those updates using two approaches:

2.1.1 Sequential Data Assimilation

Recall equation 2.2, we can also say:

$$\mathcal{P}(\mathbf{x}|\mathbf{y}) \propto \mathcal{P}(\mathbf{x}) \cdot \mathcal{P}(\mathbf{y}|\mathbf{x}) \tag{2.3}$$

This means that updates of the model state are performed as observations become available, and then, propagates the estimate. Recall the Gaussian assumption we did, the **Analysis** $\mathbf{x}_k^a$ is computed by solving the following optimization problem

$$\mathbf{x}^a = \arg\max_{\mathbf{x}} \mathcal{P}\left(\mathbf{x}|\mathbf{y}\right) , \tag{2.4}$$

with:

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^b, \mathbf{B}\right) , \tag{2.5}$$

$$\mathbf{y} \sim \mathcal{N}\left(\mathcal{H}(\mathbf{x}), \mathbf{R}\right) \tag{2.6}$$

and errors

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \mathbf{B}\right) , \tag{2.7}$$

$$\epsilon \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right) \tag{2.8}$$

Where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the background error covariance matrix, and $\mathbf{R} \in \mathbb{R}^{m \times m}$ is the observations error covariance matrix. Putting all this together:

$$\begin{aligned} \mathcal{P}(\mathbf{x}) &= \frac{1}{(2\pi)^{n/2} \cdot |\mathbf{B}|^{(1/2)}} \cdot \exp\left(-\frac{1}{2} \cdot ||\mathbf{x} - \mathbf{x}^b||^2_{\mathbf{B}^{-1}}\right) \\ &\sim \exp\left((\mathbf{x} - \mathbf{x}^b)^T \cdot \mathbf{B}^{-1} \cdot (\mathbf{x} - \mathbf{x}^b)\right) \end{aligned} \tag{2.9}$$

$$\begin{aligned} \mathcal{P}(\mathbf{y}|\mathbf{x}) &= \frac{1}{(2\pi)^{m/2} \cdot |\mathbf{R}|^{(1/2)}} \cdot \exp\left(-\frac{1}{2} \cdot ||\mathbf{y} - \mathcal{H}(\mathbf{x})||^2_{\mathbf{R}^{-1}}\right) \\ &\sim \exp\left((\mathbf{y} - \mathcal{H}(\mathbf{x}))^T \cdot \mathbf{R}^{-1} \cdot (\mathbf{y} - \mathcal{H}(\mathbf{x}))\right) \end{aligned} \tag{2.10}$$

Using equation 2.9 and equation 2.10 we can rewrite equation 2.3 as $\mathcal{P}(\mathbf{x}|\mathbf{y}) \sim \exp(-\mathcal{J}(\mathbf{x}))$ where $\mathcal{J}(\mathbf{x})$ is:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2} \cdot ||\mathbf{x} - \mathbf{x}^b||^2_{\mathbf{B}^{-1}} + \frac{1}{2} \cdot ||\mathbf{y} - \mathcal{H}(\mathbf{x})||^2_{\mathbf{R}-1} \tag{2.11}$$

So, the optimization problem in 2.4 becomes:

$$\mathbf{x}^a = \operatorname*{argmax}_{\mathbf{x}} \mathcal{P}(\mathbf{x}|\mathbf{y}) = \operatorname*{argmax}_{\mathbf{x}} \mathcal{J}(\mathbf{x}) \tag{2.12}$$

Once is solved, the posterior mode of the error distribution can be computed as follows:

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{A} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \left[\mathbf{y} - \mathbf{H} \cdot \mathbf{x}^b\right] \in \mathbb{R}^{n \times 1} \tag{2.13}$$

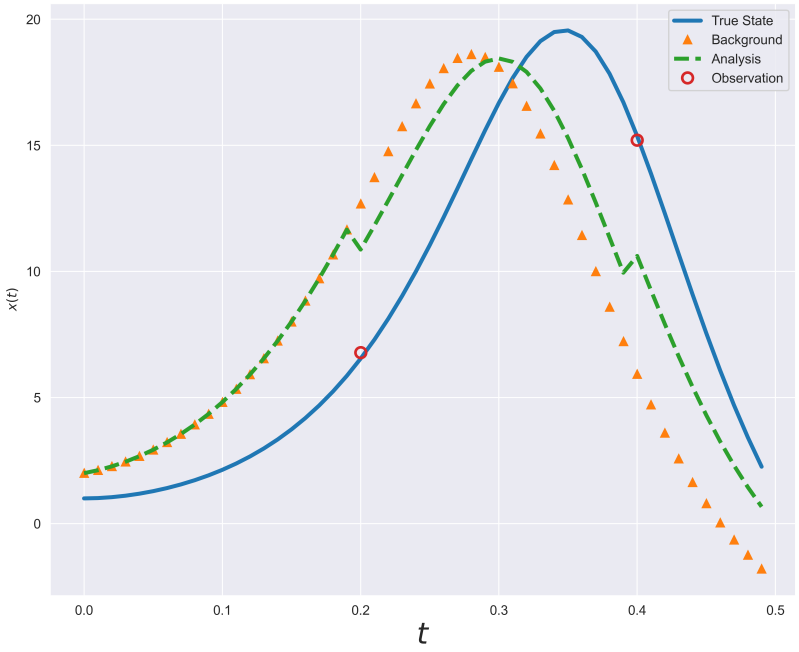


FIGURE 2.5: Sequential DA

In which $\mathcal{H}'(\mathbf{x}) \approx \mathbf{H}^T \in \mathbb{R}^{n \times m}$ is a linearized observation operator (with $\mathcal{H}(\mathbf{x}) \approx \mathcal{H}(\overline{\mathbf{x}}^b) + \mathbf{H} \cdot \left[\mathbf{x} - \overline{\mathbf{x}}^b\right]$). The analysis covariance matrix $\mathbf{A}$ reads,

$$\mathbf{A} = \left[\mathbf{B}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \in \mathbb{R}^{n \times n} \tag{2.14}$$

Solving the optimization problem given in the quadratic equation described in 2.11 is equivalent to solving the Optimal Interpolation problem in 1D (Barker et al., 2004).

This method can be seen as a Two-Step algorithm in which:

1. Propagate the system through the model to obtain a forecast
2. Once an observation is available, we update the forecast

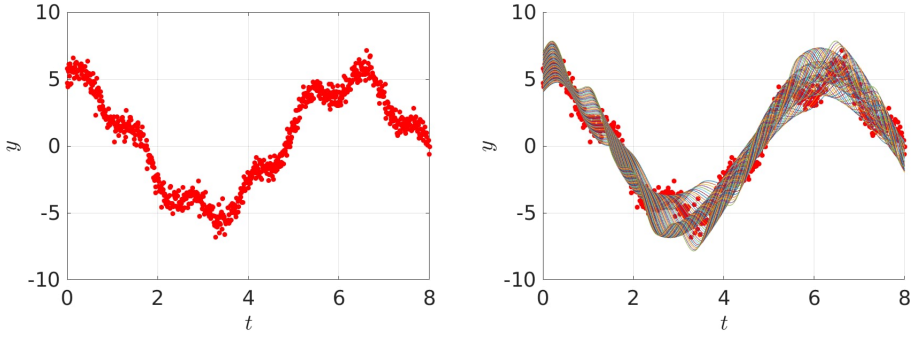
2.1.2 Variational Data Assimilation

Given multiple observations, we can assimilate them in one assimilation step. The posterior is given by

$$\mathcal{P}\left(\mathbf{x}_0 | \{\mathbf{y}_k\}_{k=0}^M\right) \propto \mathcal{P}(\mathbf{x}_0) \cdot \prod_{k=0}^{M} \mathcal{L}(\mathbf{x}_k | \mathbf{y}_k) \tag{2.15}$$

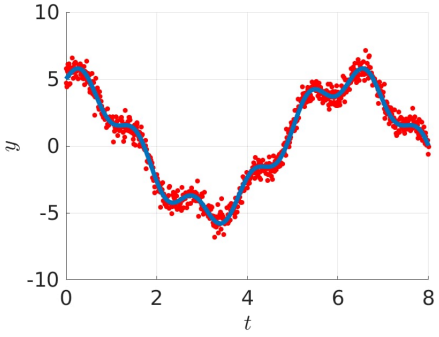
And the optimization process is performed in a single shot.

$$\mathbf{x}_0^a = \arg\max_{\mathbf{x}_0} \mathcal{P}\left(\mathbf{x}_0 | \{\mathbf{y}_k\}_{k=0}^M\right) \tag{2.16}$$



(A) Observations

(B) Forecasts



(C) Analysis

We can write $\mathcal{J}(\mathbf{x})$ as (Lorenc, 2003b; Lorenc, 2003a):

$$\mathcal{J}(\mathbf{x}_0) = \left\|\mathbf{x}_0 - \mathbf{x}_0^b\right\|^2_{\mathbf{B}_0^{-1}} + \sum_{k=0}^{M} \left\|\mathbf{y}_k - \mathcal{H}_k\left(\mathbf{x}_k\right)\right\|^2_{\mathbf{R}_k^{-1}}, \tag{2.17}$$

To solve this optimization problem, we require the use of Adjoints which are computationally expensive to compute and complex to calculate, even more, to validate.

2.2 The Ensemble Kalman Filter

The previous section defined what is called 3DVAR and 4DVAR, but there is an issue remaining to fully compute the updates of the forecast. Given a model $\mathcal{M}$ which has 10^6 components, if we want to compute $\mathbf{B}$ that would give us a complexity, in the worst case, of $\mathcal{O}(n^3)$ and memory space of 8Tb. We can see, as the model grows in size, is more infeasible the explicit computation of this matrix (Pourahmadi, 2011; Fan, Liao, and Liu, 2016). Even more, to solve equation 2.16 we require the use of Adjoints and the calculation of the Tangent model, which are labor-intense and computational expensive as can be seen in (Gustafsson and Bojarova, 2014; Stengel et al., 2009). So, to deal with this, let's consider N model realizations of an initial state with a small perturbation, which will be called ***ensemble members***, and we *estimate* the Covariance matrix and its inverse (also called the Precision matrix) using them. Using such an ensemble the is no requirement to solve the Tangent model or using Adjoints (Kalnay, 2002). In this proposal, we will be using the so-called **Ensemble Kalman Filter** (EnKF from now on). In the EnKF context, an ensemble of model realizations is utilized, (Houtekamer and Mitchell, 1998; Stroud, Katzfuss, and Wikle, 2018).

The Kalman filter, first proposed in (Kalman, 1960; Kalman and Bucy, 1961) as a result of the control theory, provides a mathematical description of the assimilation problem but suffers from drawbacks the Ensemble Kalman Filter (EnKF) tries to solve (Evensen, 1994; Ito et al., 2016). As before, the Gaussianity assumption must hold, to be significant the DA process, and the prediction is the mean of the posterior and the moments of the Gaussian PDF are estimated from ensembles of small size respect the state space dimensions (Nino Ruiz, Sandu, and Anderson, 2014; Godinez and Moulton, 2012)

The EnKF is a sequential Monte Carlo method for parameter and state estimation of highly non-linear models (Evensen, 2006). The main idea behind these approaches is to build ensemble sub-spaces (by propagating the ensembles) where analysis increments can be estimated (Wang et al., 2007). The popularity of the EnKF comes from its simple formulation and relatively easy implementation (Lorenc, 2003b; Gillijns et al., 2006). In the EnKF, an ensemble of N model realizations,

$$\mathbf{X}^b = \left[\mathbf{x}^{b[1]}, \mathbf{x}^{b[2]}, \ldots, \mathbf{x}^{b[N]}\right] \in \mathbb{R}^{n\times N}, \tag{2.18}$$

where $\mathbf{x}_k^{b[i]} \in \mathbb{R}^{n\times 1}$ refers for the i-th ensemble member, for $1 \leq i \leq N$, at time k, for $0 \leq k \leq M$. We use this ensemble to estimate the prior error distribution,

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^b, \mathbf{B}\right)$$

via the via the empirical moments of the ensemble, in which the ensemble mean is:

$$\mathbf{x}^b \approx \overline{\mathbf{x}}^b = \frac{1}{N}\cdot\sum_{e=1}^{N}\mathbf{x}^{b[e]} \in \mathbb{R}^{n\times 1}, \tag{2.19}$$

and

$$\mathbf{B} \approx \mathbf{P}^b = \frac{1}{N-1}\cdot\mathbf{\Delta X}\cdot\mathbf{\Delta X}^T \in \mathbb{R}^{n\times n}, \tag{2.20}$$

where $\mathbf{\Delta X} \in \mathbb{R}^{n\times N}$ is the matrix of member deviations and is given by,

$$\mathbf{\Delta X} = \mathbf{X}^b - \overline{\mathbf{x}}^b\cdot\mathbf{1}_N^T. \tag{2.21}$$

The assimilation step is performed onto the space spanned by the ensemble members (since this is all info we have from the numerical model). Any element in the ensemble space can be written as follows:

$$\mathbf{x} = \mathbf{x}^b + \mathbf{\Delta X}\cdot\alpha, \tag{2.22}$$

This formulation is equivalent to

$$\mathbf{x} - \overline{\mathbf{x}}^b \in \mathbf{range}\left\{\mathbf{\Delta X}\right\} \approx \mathbf{range}\left\{\mathbf{B}^{1/2}\right\}$$

The assimilation process can be performed, as follows:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{K}\cdot\mathbf{D} \in \mathbb{R}^{n\times N}, \tag{2.23}$$

Where $\mathbf{K}$ is known as the Kalman Gain and reads

$$\mathbf{K} = \mathbf{P}^b\cdot\mathbf{H}^T\cdot\mathbf{C}^{-1},$$

$\mathbf{C}$ is known as the Innovation Matrix,

$$\mathbf{C} = \left[\mathbf{R} + \mathbf{H} \cdot \mathbf{P}^b \cdot \mathbf{H}^T\right] ,$$

and $\mathbf{D} \in \mathbb{R}^{m \times N}$ is the matrix of innovations on the synthetic observations:

$$\mathbf{D} = \mathbf{y} \cdot \mathbf{1}_N^T - \mathbf{H} \cdot \mathbf{X}^b + \mathbf{E} , \tag{2.24}$$

where the columns of $\mathbf{E} \in \mathbb{R}^{m \times N}$ are samples from a zero-mean Normal distribution with data-error covariance matrix $\mathbf{R} \in \mathbb{R}^{m \times m}$ (Dovera and Della Rossa, 2011; Abaza et al., 2017).

Alternative formulations are:

$$\mathbf{X}^a = \mathbf{X}^b + \left[\mathbf{B}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{D}$$

$$\mathbf{X}^a = \left[\mathbf{B}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \cdot \left[\mathbf{B}^{-1} \cdot \mathbf{X}^b + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{Y}\right]$$

The analysis covariance matrix $\mathbf{A}$ reads,

$$\mathbf{A} = \left[\mathbf{B}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \in \mathbb{R}^{n \times n} ,$$

The observations $\mathbf{Y} \in \mathbb{R}^{m \times N}$ are in the form of

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, ..., \mathbf{y}_N]$$
$$\mathbf{y}_i = \mathbf{y}^o + \boldsymbol{v_i},\ \boldsymbol{v_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) ,$$

which means they are a random variable (Dovera and Della Rossa, 2011; Abaza et al., 2017) in which $\mathbf{y}^o \in \mathbf{R}^{m \times 1}$ is the real observation and every other are perturbed observations.

An efficient way to compute those updates is as follows,

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{Z} \in \mathbb{R}^{n \times N} , \tag{2.25}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times N}$ can be obtained by the solution of the linear system of equations,

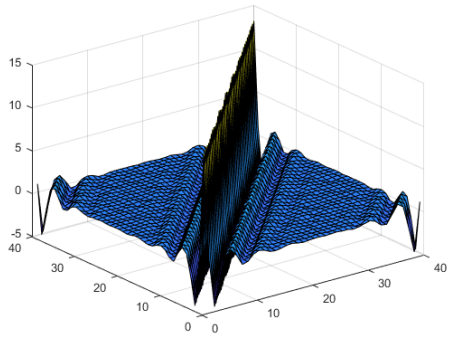
$$\left[[\mathbf{B}]^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right] \cdot \mathbf{Z} = \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{\Delta Y} \tag{2.26}$$

2.2.1 Filter Drawbacks

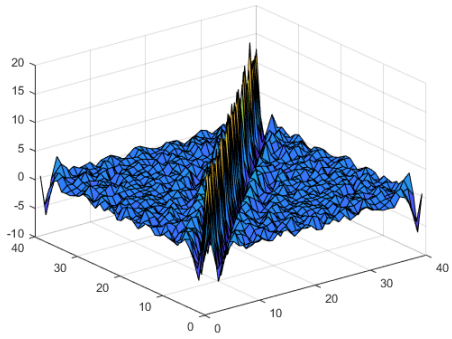
As we previously mentioned, in a naive approach, the computation of $\mathbf{B}$ will require $\mathcal{O}(n^3)$ while the EnKF requires $\mathcal{O}(n^2 \cdot N)$ operations where $N << n$, as for the computation of the gain requires $\mathcal{O}(n^2 \cdot m)$, and the EnKF requires $\mathcal{O}(n \cdot m \cdot N)$ operations (Roth et al., 2017). A more detailed explanation can be found in (Mandel, 2006). Since the hundreds bound ensemble sizes while model resolutions are by the millions, sampling errors impact the quality of analysis innovations, and as a direct consequence, we can poorly estimate posterior errors (Houtekamer and Mitchell, 1998; Anderson, 2001; Buehner, 2011; Kondo and Miyoshi, 2016)

To avoid this issue, we will address two main strategies.

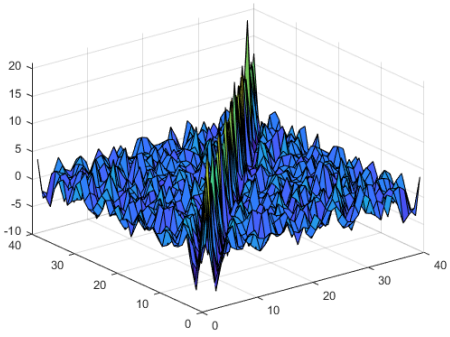
- Localization
- Inflation



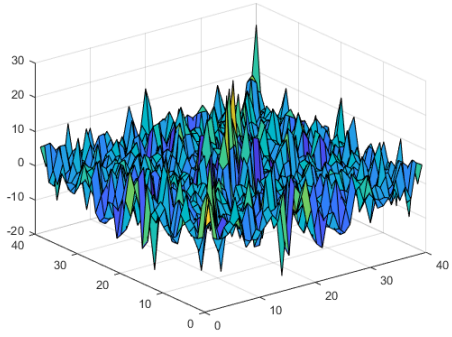
(D) $N = 10^5$



(E) $N = 500$



(F) $N = 60$



(G) $N = 6$

FIGURE 2.6: Covariance Matrix estimations

Localization

To reduce the effects of the sampling noise, *localization* techniques are proposed, such methods deal with the spurious correlations that might develop in the assimilation process. (Houtekamer and Mitchell, 1998; Anderson, 2001; Buehner, 2011; Kondo and Miyoshi, 2016). These techniques rely on localization functions that define the localization length. In these methods, for each model component $1 \leq i \leq n$, a neighborhood $P(i, \delta)$ is defined based on some labeling of model components and a radius of influence $r \in \mathbb{R}^+$ (Nino-Ruiz, Sandu, and Deng, 2018), therefore,

$$j \in P(i, r) \Leftrightarrow d\left(x_i, x_j\right)^2 \leq r^2, \text{and } j < i\,, \tag{2.27}$$

where $d(\bullet, \bullet)$ denotes a distance function. In practice, this labeling can be done in many different ways, the most common being row-major and column-major. This definition is applied to discard long distanced components that are not related, by the model assumptions. This enforces a sparse matrix structure of the estimated covariance matrix. This technique relies on some prior knowledge of the structure of covariance. For example, the geographical distance could lead us to believe distant components are uncorrelated.

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

FIGURE 2.7: Labelling

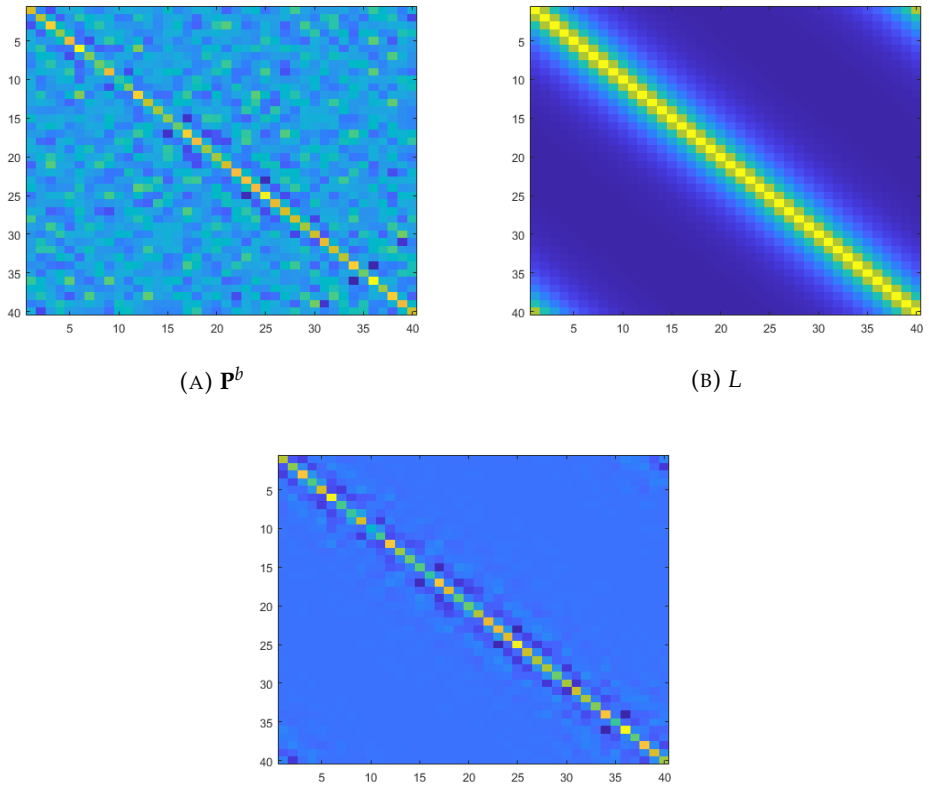
One traditional way to achieve this is by using the Schur product denoted by $\circ$. Given the matrix $\widehat{\mathbf{\Lambda}}$, which is defined by

$$\left\{\widehat{\mathbf{\Lambda}}\right\}_{i,j} = \exp\left(-\frac{1}{2}\cdot\frac{d\,(i,\,j)^2}{r^2}\right)\,,\ \text{for}\ 1 \leq i,j \leq n\,, \tag{2.28}$$

we can localize $\mathbf{P}^b$ by applying

$$\widetilde{\mathbf{P}}^b = \widehat{\mathbf{\Lambda}} \circ \mathbf{P}^b \in \mathbb{R}^{n\times n}\,, \tag{2.29}$$

The distance function usually is the Euclidean distance. This approach filters long-range spurious correlations.



(A) $\mathbf{P}^b$ (B) L

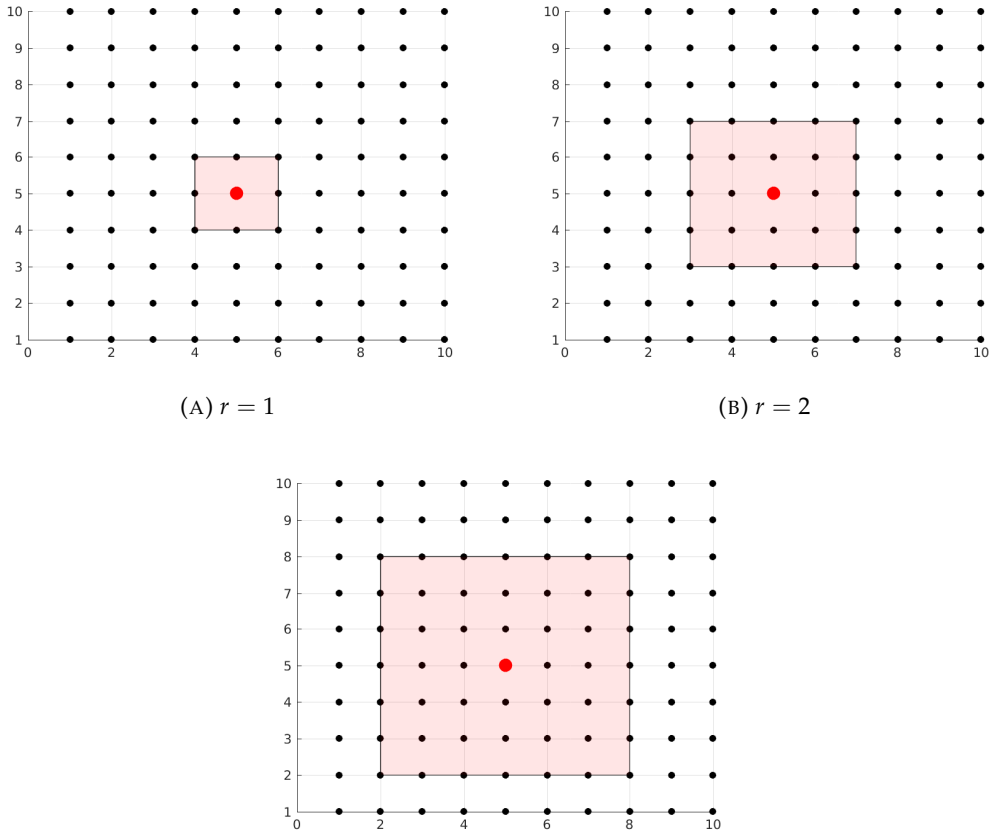
(C) $\widetilde{\mathbf{P}}^b$

FIGURE 2.8: Localization

The correct selection of the radius of influence is the core in this technique, as a wide range influence would mean we maintain those spurious correlations, but if it's too short, then dynamic correlations might be lost. In practice, localization methods are commonly used to artificially increase the rank of $\mathbf{P}^b$ and to mitigate

the impact of spurious correlations during the analysis steps (Hamill, Whitaker, and Snyder, 2001). These localization methods can be: covariance matrix localization (**B**-localization) (Lei, Whitaker, and Bishop, 2018), domain localization, and observation localization (**R**-localization) (Anderson, 2001; Han, Zhang, and Sun, 2018; Anderson, 2019).

We can also think in a spatial localization in which each component is surrounded by a local box, and for each local box, we perform local assimilation with all the information contained in there, then mapped back to the global domain where the analysis is performed. A 2D example can be seen in 2.9 where the red dot is the model component and the red square are the Neighbors' components.



(A) $r = 1$ (B) $r = 2$

(C) $r = 3$

FIGURE 2.9: Local domains for different radii of influence r.

The local ensemble Kalman filter (LEnKF) (Ott et al., 2004b; Tong, 2018) exploits local information via a radius length δ to mitigate the impact of spurious correlations. It is equivalent to apply the EnKF equations (2.23) to each model component within its corresponding local box.

Other deterministic methods such as the local ensemble transform Kalman filter (LETKF). The Local Ensemble Transform Kalman Filter, a deterministic formulation of the EnKF works using this localization technique and has been widely used in operational data assimilation centers (Hunt, Kostelich, and Szunyogh, 2007; Tippett et al., 2003). The mean of the analysis distribution is estimated in the ensemble space as follows, avoid the use of synthetic data (2.24) to avoid inducing more sampling error during assimilation steps. In the global formulation of the LETKF, the posterior

mode is computed as follows:

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{\Delta X}^b \cdot \left[(N-1)\cdot \mathbf{I} + \mathbf{Q}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{Q}\right]^{-1} \cdot \mathbf{Q}^T \cdot \mathbf{R}^{-1} \cdot \left[\mathbf{y} - \mathbf{H} \cdot \overline{\mathbf{x}}^b\right] \in \mathbb{R}^{n\times 1}, \tag{2.30a}$$

where $\mathbf{Q} = \mathbf{H} \cdot \mathbf{\Delta X} \in \mathbb{R}^{m\times N}$. The posterior ensemble is then built about (2.30a) as follows:

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \cdot \mathbf{1}^T + \mathbf{\Delta X} \cdot \left[(N-1)\cdot \mathbf{I} + \mathbf{Q}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{Q}\right]^{-1/2}. \tag{2.30b}$$

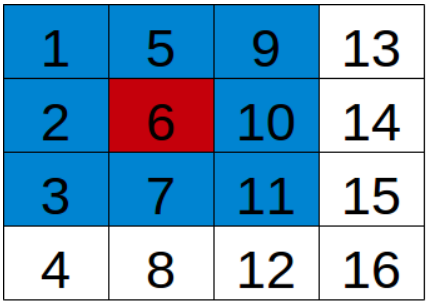
Similar to the LEnKF, the set of equations (2.30) is applied to each model component within its local box.

The assimilation step is applied to each model component for a given radius of influence r, with which the global analysis state will be obtained. Since the most expensive computation is the inversion of $\widetilde{\mathbf{P}^a}$, the computational effort of the LETKF reads,

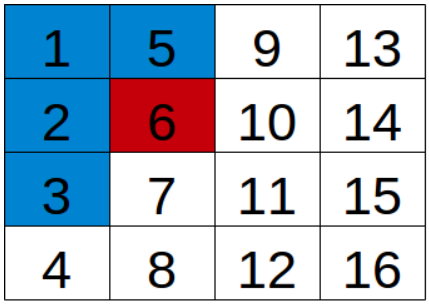
$$\mathcal{O}\left(\varphi \cdot n \cdot N^3\right),$$

where φ denotes the local box sizes.

Another efficient implementation is an Ensemble Kalman Filter based on a Modified Cholesky decomposition (EnKF-MC). This implementation uses the concept of spatial predecessors to obtain sparse estimators of precision matrices (Levina, Rothman, Zhu, et al., 2008). The predecessors of model component i, from now on $\Pi(i,\, r)$, for $1 \leq i \leq n$ and a radius of influence $r \in \mathbb{Z}^+$, are given by the set of components whose labels are lesser than that of the i-th one. As example, if we define $r = 1$ and $i = 6$, using column-major.



(A) Local box $r = 1, i = 6$.



(B) Predecessors $r = 1, i = 6$

This idea is exploited in the EnKF formulation proposed in (Nino-Ruiz, Sandu, and Deng, 2017; Nino-Ruiz, Sandu, and Deng, 2018) wherein the following estimator is employed to approximate the precision matrix (Bickel, Levina, et al., 2008)

$$\widehat{\mathbf{B}}^{-1} = \mathbf{L}^T \cdot \mathbf{D}^{-1} \cdot \mathbf{L} \in \mathbb{R}^{n\times n}, \tag{2.31}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is a unitary lower-triangular matrix, and $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix.

where the Cholesky factor $\mathbf{L}_k \in \mathbb{R}^{n\times n}$ is a lower triangular matrix,

$$\{\mathbf{L}_k\}_{i,v} = \begin{cases} -\beta_{i,v,k} & , v \in P(i,r) \\ 1 & , i = v \\ 0 & , othercase \end{cases} , \tag{2.32}$$

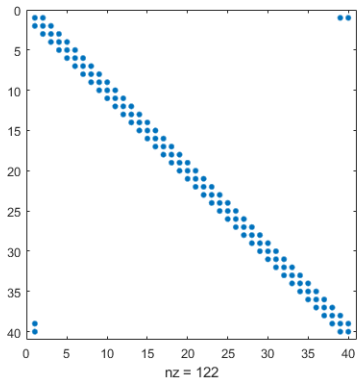
whose strict lower triangular elements $\beta_{i,v,k}$ are obtained by fitting the parameters of different models given by,

$$\mathbf{x}^T_{[i]_k} = \sum_{v \in P(i,\delta)} \beta_{i,v,k} \cdot \mathbf{x}^T_{[v]_k} + \gamma_{ik} \in \mathbb{R}^{N \times 1} , \; 1 \leq i \leq n , \tag{2.33}$$

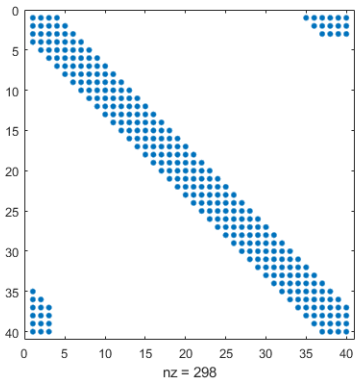
where $\mathbf{x}^T_{[i]_k} \in \mathbb{R}^{N \times 1}$ denotes the i-th row (model component) of the ensemble, components of vector $\gamma_{ik} \in \mathbb{R}^{N \times 1}$ are samples from a Normal distribution with zero mean and unknown variance σ_k^2, and $\mathbf{D}_k \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal elements are as follows,

$$\begin{aligned} \{\mathbf{D}_k\}_{i,i} &= \widehat{\mathbf{var}} \left(\mathbf{x}^T_{[i]_k} - \sum_{v \in P(i,\delta)} \beta_{i,v,k} \cdot \mathbf{x}^T_{[j]_k} \right)^{-1} \\ &\approx \mathbf{var} \left(\gamma_{ik} \right)^{-1} = \frac{1}{\sigma_k^2} > 0 , \\ \text{with} \quad \{\mathbf{D}_k\}_{1,1} &= \widehat{\mathbf{var}} \left(\mathbf{x}^T_{[1]_k} \right)^{-1} , \end{aligned}$$

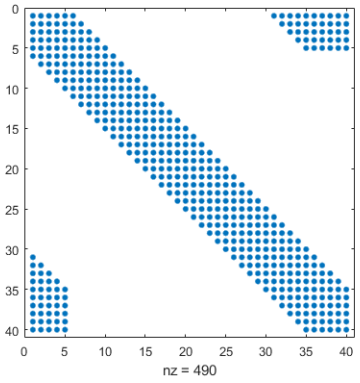
where $\mathbf{var}(\bullet)$ and $\widehat{\mathbf{var}}(\bullet)$ denote the actual and the empirical variances, respectively. In $\mathbf{D}_k$ are stored the residuals of the regression model we used. The structure of $\mathbf{L}$ can be sparse, which implies huge savings in terms of memory usage under current operational data assimilation settings wherein n can be very large. Besides, $\widehat{\mathbf{B}}^{-1}$ can be represented in terms of his Cholesky factors, and therefore, efficient manners to compute the ensemble can be derived Nino Ruiz, Sandu, and Anderson, 2014.



(C) $r = 1$



(D) $r = 3$



(E) $r = 5$

FIGURE 2.10: Structure of $\widehat{\mathbf{B}}^{-1}$ by the Cholesky factors

In this context, the posterior ensemble is computed as follows:

$$\mathbf{X}^a = \mathbf{X}^b + \widehat{\mathbf{A}} \cdot \mathbf{\Delta Y} \in \mathbb{R}^{n \times N}, \tag{2.34}$$

where the analysis error covariance matrix reads,

$$\widehat{\mathbf{A}} = \left[\widehat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \in \mathbb{R}^{n \times n}, \tag{2.35}$$

Using algebraic identities, we can obtain,

$$\widehat{\mathbf{B}} = \mathbf{L}^{-1} \cdot \mathbf{D} \cdot \mathbf{L}^{-T} \in \mathbb{R}^{n \times n},$$

The structure of both $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{B}}^{-1}$ is strictly related to the sparse structure of $\mathbf{L}$, so we can take advantage to obtain both without their explicit computation, which is prohibitive in a higher-dimensional context.

Inflation

While localization methods reduce the impact of spurious correlations, covariance inflation mitigates the impact of underestimation sample variances (Westgate, 2016; Lei and Whitaker, 2017; Lee, Majda, and Qi, 2016; Putnam et al., 2017). Firstly introduced in (Anderson and Anderson, 1999), typically, ensemble members are inflated before to the forecast step to enrich the background error information for the next

assimilation cycle and to reduce the odds of ensemble collapsing. For instance, after the assimilation step of EnKF, the ensemble members (2.25) are inflated by a factor of $\rho > 1$ about the analysis mean,

$$\mathbf{X}^{a,\rho} = \overline{\mathbf{x}}^a \cdot \mathbf{1}^T + \rho \cdot \mathbf{\Delta X}^a \,,$$

where $\mathbf{\Delta X}^a = \mathbf{X}^a - \overline{\mathbf{x}}^a \cdot \mathbf{1}^T \in \mathbb{R}^{n \times N}$ are the innovations about the analysis mean. Thus, the (co) variances in $\mathbf{P}^a$ are inflated by a factor of ρ^2. The inflation factor r is normally chosen to be slightly greater than 1. The inflation can vary depending on the size of the ensemble and tends to improve the accuracy of the filter by increasing the variance of the samples artificially.

In both techniques, localization and inflation are assumed to be maintained throughout the assimilation process, but there have been approaches for an adaptive value for both of them. Even more, the task of choosing an optimal radio or inflation value is complex, so in operative cases is obtained heuristically.

Chapter 3

Methods and Objectives

3.1 Problem Statement

With the mentioned researches, we can mention the following issues:

3.1.1 Limitations

- Ensemble-based methods are highly sensitive to sampling noise.
- Meteorological Simulations demand high computational efforts.
- Real datasets can be found in specific resolutions.
- Models' configurations can be too complex to vary their settings to perform experiments.

3.1.2 Operational

- No real-time is accessible in Colombia. Data is commonly obtained by requests to governmental agencies (INVEMAR e IDEAM), and then it is available weeks later.
- There are no numerical models to estimate wind components in Colombia (and to exploit our knowledge about our ecosystems).

3.2 Expected Main Contributions

In summary, the main contribution we hope to achieve with our proposal is:

- to develop a data assimilation library that could help to test new methods and already tested ones.
- to develop an implementation of the EnKF-MC based on the observations.
- to implement a straightforward way to modify our numerical model.
- to develop an interpolation tool that could resample observations obtained from the NOAA to specific grid resolutions.

3.3 Objectives

3.3.1 Main Objective

To design and implement efficient formulations of ensemble-based methods for sequential data assimilation.

3.3.2 Specific Objectives

1. To design and implement efficient formulations of ensemble-based methods.
2. To design and implement a data assimilation toolbox for AT-GCM models.
3. To validate the formulations via metrics from the specialized literature.

3.4 Methodology

In this section, we briefly describe the steps to accomplish the objectives in Section 3.3.

1. Design and Implement a package that allow us to create instances of the SPEEDY model at ease, with high versatility in configuration and settings.
2. Build ensembles of model realizations via an the SPEEDY model for different configurations.
3. Design and Implement efficient formulations of literature-known Data Assimilation method as are the LETKF, LEnKF.
4. Implement a version of the EnKF-MC onto the observations space.
5. Deploy the designed solution into a High Performance Computing environment to run experiments.

All the implementations are made using the scientific computational language Python, and modifiying the Fortran configuration of the model.

Chapter 4

Proposed Method

4.1 An Modified Cholesky implementation of the EnKF: An observation space version

Before we start, we assume that, the observational operator is nearly linear and/or it can be easily applied (Sakov, Evensen, and Bertino, 2010), the data error covariance matrix possesses a simple structure and/or it can be easily decomposed (Evensen, 2009), and that observational networks are sparse (Anderson, 2001).

We know it is possible to obtain an approximation of the precision analysis covariance matrix in terms of sparse Cholesky factors (Nino-Ruiz, Sandu, and Deng, 2016) which can be exploited during the assimilation step (2.34) to avoid the explicit solution of the linear system.

The assimilation process, for instance, can be stochastically performed,

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{B}^b \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \mathbf{B}^b \cdot \mathbf{H}^T\right]^{-1} \cdot \mathbf{D} \in \mathbb{R}^{n \times N}$$

We can use a Cholesky Decomposition to estimate $\widehat{\mathbf{B}}^{-1}$ by

$$\widehat{\mathbf{B}}^{-1} = \widetilde{\mathbf{L}}_k^T \cdot \widetilde{\mathbf{D}}_k^{-1} \cdot \widetilde{\mathbf{L}}_k = \widetilde{\mathbf{X}}_k^T \cdot \widetilde{\mathbf{X}}_k \,, \tag{4.1}$$

so, the analysis is given by

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}_k^b + [\widehat{\mathbf{B}}_k]^{-1} \cdot \widetilde{\mathbf{H}}_k^T \cdot \left[\mathbf{R}_k + \widetilde{\mathbf{H}}_k \cdot [\widehat{\mathbf{B}}_k]^{-1} \cdot \widetilde{\mathbf{H}}_k^T\right]^{-1} \cdot \mathbf{D} \in \mathbb{R}^{n \times N} \,, \tag{4.2}$$

replacing equation 4.1 into equation 4.2

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}_k^b + [\widetilde{\mathbf{X}}_k^T \cdot \widetilde{\mathbf{X}}_k]^{-1} \cdot \widetilde{\mathbf{H}}_k^T \cdot \left[\mathbf{R}_k + \widetilde{\mathbf{H}}_k \cdot [\widetilde{\mathbf{X}}_k^T \cdot \widetilde{\mathbf{X}}_k]^{-1} \cdot \widetilde{\mathbf{H}}_k^T\right]^{-1} \cdot \mathbf{D} \in \mathbb{R}^{n \times N}$$

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}_k^b + \widetilde{\mathbf{X}}_k^{-1} \cdot \underbrace{\widetilde{\mathbf{X}}_k^{-T} \cdot \widetilde{\mathbf{H}}_k^T}_{\mathbf{Z_k}} \cdot \left[\mathbf{R}_k + \widetilde{\mathbf{H}}_k \cdot \widetilde{\mathbf{X}}_k^{-1} \cdot \underbrace{\widetilde{\mathbf{X}}_k^{-T} \cdot \widetilde{\mathbf{H}}_k^T}_{\mathbf{Z_k}}\right]^{-1} \cdot \mathbf{D} \in \mathbb{R}^{n \times N}$$

Let $\widetilde{\mathbf{H}}_k = \mathbf{Z_k} \cdot \widehat{\mathbf{X}}_\mathbf{k}^\mathbf{T}$, then

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}_k^b + \widetilde{\mathbf{X}}_k^{-1} \cdot \mathbf{Z}_\mathbf{k}^\mathbf{T} \cdot \underbrace{\left[\mathbf{R}_k + \mathbf{Z_k} \cdot \mathbf{Z}_\mathbf{k}^\mathbf{T}\right]^{-1} \cdot \mathbf{D}}_{\mathbf{Q_k}} \in \mathbb{R}^{n \times N} \,, \tag{4.3}$$

where $\mathbf{Q_k}$ in equation 4.3 can be efficiently computed by $\left[\mathbf{R}_k + \mathbf{Z_k} \cdot \mathbf{Z}_\mathbf{k}^\mathbf{T}\right] \cdot \mathbf{Q_k} = \mathbf{D}$,

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}_k^b + \underbrace{\widetilde{\mathbf{X}}_k^{-1} \cdot \mathbf{Z}_\mathbf{k}^\mathbf{T} \cdot \mathbf{Q}_\mathbf{k}}_{\widetilde{\mathbf{\Delta X}}} \in \mathbb{R}^{n \times N} ,$$

similarly, solving $\widetilde{\mathbf{X}}_k \cdot \widetilde{\mathbf{\Delta X}}_k = \mathbf{Z}_\mathbf{k}^\mathbf{T} \cdot \mathbf{Q}_\mathbf{k}$ we obtain,

$$\widetilde{\mathbf{X}}_k^a = \mathbf{X}^b + \widetilde{\mathbf{\Delta X}}_k$$

where $\widetilde{\mathbf{H}}_k = [\mathbf{H}_k, \mathbf{0}] \in \mathbf{R}$.

In this method, the update is performed for all the grids instead of local updates for each segment.

In our proposal, we begin by choosing a numerical model which mimics the dynamics of weather. For this purpose, numerical models such as the Atmospheric General Circulation Model (ATGCM SPEEDY) (Amezcua, Kalnay, and Williams, 2011) can be employed. Once the numerical model is chosen, snapshots of an ensemble of model realizations (2.18) are taken at $M+1$ observation times. At step k, for $0 \leq k \leq M$, the background ensemble $\mathbf{X}_k^b$ (2.18) is employed to estimate a full-rank square-root approximation of the precision matrix of background errors $\mathbf{B}_k^{-1}$ via a modified Cholesky decomposition (2.31):

$$\widehat{\mathbf{B}}_k^{-1/2} = \widehat{\mathbf{L}}_k^T \cdot \widehat{\mathbf{D}}_k^{-1/2} \in \mathbb{R}^{n \times n} . \tag{4.4}$$

At this step, we choose a radius of influence (localization radius) r to compute the factor $\widehat{\mathbf{L}}_k^T$. Beyond the scope of this radius (and the predecessors of model components), all components of $\widehat{\mathbf{L}}_k^T$ are assumed zero. As previously mentioned in section 2.2.1 we exploit the fact that, when the error correlations of two model components are conditionally independent (for a given r), their corresponding entry in the precision matrix of background errors is zero. This results in a sparse Cholesky factor $\widehat{\mathbf{L}}_k^T$ and even more, a localized square-root precision matrix. In this manner, the impact of sampling errors can be mitigated in the square-root approximations (4.4).

After the assimilation step, the ensemble members are inflated by a factor of $\rho > 1$ about the analysis mean,

$$\mathbf{X}^{a,\rho} = \overline{\mathbf{x}}^a \cdot \mathbf{1}^T + \rho \cdot \mathbf{\Delta X}^a ,$$

where $\mathbf{\Delta X}^a = \mathbf{X}^a - \overline{\mathbf{x}}^a \cdot \mathbf{1}^T \in \mathbb{R}^{n \times N}$ are the innovations about the analysis mean. Thus, the covariances in $\widehat{\mathbf{B}}^a$ are inflated by a factor of ρ^2.

Note that, β in equation 2.33 is the solution of the optimization problem

$$\beta_i = \underset{\beta}{\operatorname{argmin}} \, \|\mathbf{x} - \mathbf{Z}_i \cdot \beta\|_2^2 \tag{4.5}$$

where each column of $\mathbf{Z}$ is the i-th model component. Since the ensemble can be smaller than the number of model components, $\mathbf{Z}$ can be rank deficient, so regularization techniques can be used to overcome this situation. In this proposal, we will be decomposing $\mathbf{Z}$ using singular value decomposition (SVD), so we obtain:

$$\mathbf{Z}_i = \mathbf{U}^{\mathbf{Z}_i} \cdot \mathbf{\Sigma}^{\mathbf{Z}_i} \cdot \mathbf{V}^{\mathbf{Z}_i T}$$

where $\mathbf{U}^{\mathbf{Z}_i} \in \mathbb{R}^{p_i \times p_i}$ and $\mathbf{V}^{\mathbf{Z}_i T} \in \mathbb{R}^{p_i \times p_i}$ are the right and left singular vectors of $\mathbf{Z}_i$ and p_i are the predecessors of the i-th component. Likewise, $\mathbf{\Sigma}^{\mathbf{Z}_i} \in \mathbb{R}^{N \times N}$ is a diagonal matrix whose diagonal entries are the singular values of $\mathbf{Z}_i$ in descending order. The solution of 4.5 can be computed as follows

$$\beta_i = \sum_{j=i}^{k_i} \frac{1}{\tau_j} \cdot \mathbf{u}_j^{\mathbf{Z}_i} \cdot \mathbf{v}_j^{\mathbf{Z}_i T} \cdot \mathbf{x}_i \text{ with} \frac{\tau_j}{\tau_{max}} > \sigma_r$$

where τ_j is the j-th singular value, with their corresponding left and right singular vectors, and σ is a defined threshold, which neglects smaller singular values which are more sensitive to the noise in $\mathbf{x}_i$

Regarding the localization process, is worth mentioning the SPEEDY model contains 8 layers of pressure for each variable. Taking this into consideration, we use three "localization boxes":

- A traditional Local Box for a variable and each variable is assimilated individually. We perform localization on each component on each layer a perform the assimilation as traditional sense.

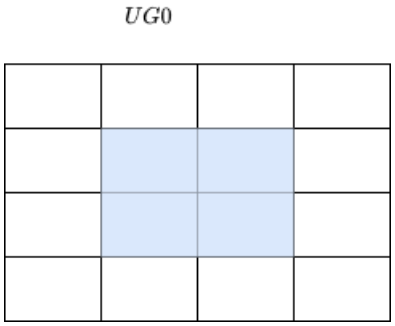


FIGURE 4.1: Traditional

- A coupled box, in which we correlate the localized components on different variables, and assimilated at the same time. This means we take into consideration the behavior and relation among variables in the localization and assimilation process.

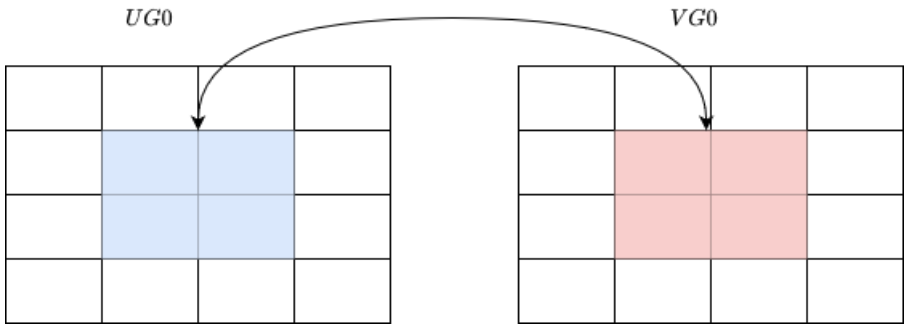


FIGURE 4.2: Coupled localization

- A level localization in which we relate model components in different levels, so in the assimilation process we assimilate them at time.

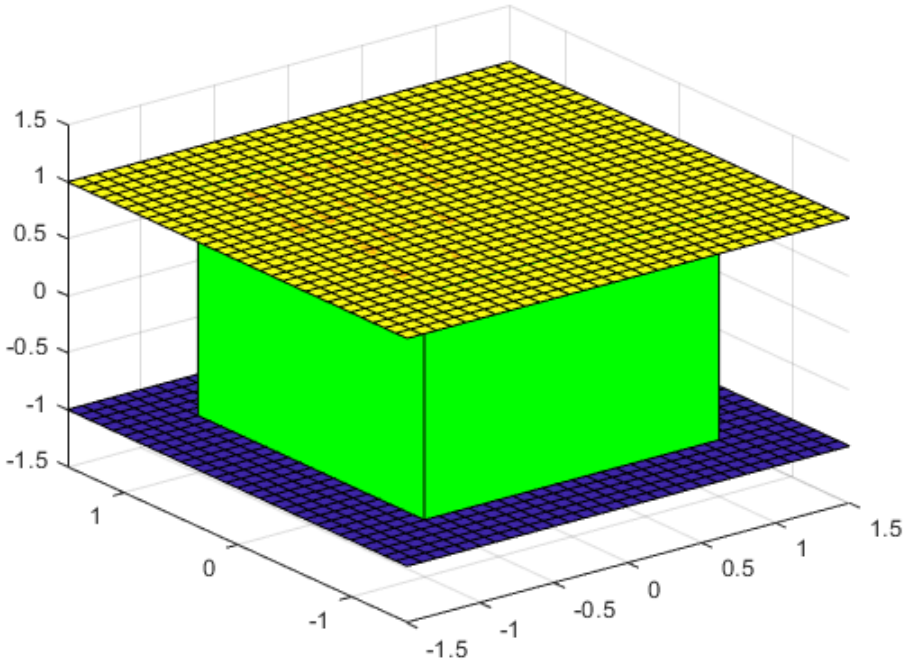


FIGURE 4.3: Localization by levels for a given variable

This offers versatility in the assimilation process which can be beneficial for the model under consideration, and many other physical models, as allows to model coupled behaviors in the phenomena. Once the assimilation process is completed, each block is computed back to the complete scheme and propagated.

4.2 Leaping Propagation

The set of differential equations that model the dynamical systems are commonly solved by numerical means, using a numerical integrator like Runge-Kutta or finite differences and solving:

$$\frac{du}{dx} = F(u, x)$$

But in these models, we commonly have a mixture on the right-hand side, so we cannot compute it explicitly in our numerical integrator. In this context, is commonly used a leaping strategy which will allow us to compute the evolution of our system, by using an intermediate step as shown in figure 4.4

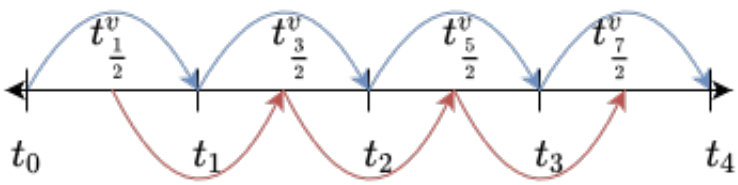


FIGURE 4.4: Leapfrog

The appealing of this method is its simplicity, as it is almost identical to the traditional Euler method, but is second order. Besides, it conserves the energy as well as the angular momentum of the dynamical process and is time-reversal invariant.

Having these initial conditions (the initial condition itself and the intermediate step), the evolution of the system is as figure 4.5

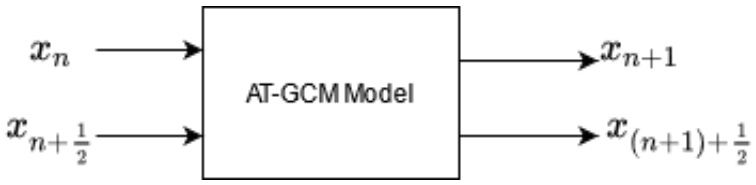


FIGURE 4.5: Numerical model evolution

A similar procedure can be applied to higher-order integrators like the Yoshida integrator. In a general sense, models which uses a leaping strategies take an initial state x_i and generate the leaping step$x_{i+\frac{1}{2}}$, and use them to propagate the model. The data assimilation process is performed at time $i+1$ when observations are obtained, forecast and leaping step updates are computed, and propagate into those creating the new state. This process is repeated in a given time window. In our proposal, we are going to update only the forecast state using the observations and let the model propagate the update to the intermediate step.

Given the two steps of the leapfrog integration

$$\mathbf{X}_0^b = \left[\mathbf{x}_0^{b[1]}, \mathbf{x}_0^{b[2]}, \ldots, \mathbf{x}_0^{b[N]}\right] \in \mathbb{R}^{n\times N}$$

and

$$\mathbf{X}_1^b = \left[\mathbf{x}_1^{b[1]}, \mathbf{x}_1^{b[2]}, \ldots, \mathbf{x}_1^{b[N]}\right] \in \mathbb{R}^{n\times N},$$

where $\mathbf{X}_0^b$ is the forecasted state and $\mathbf{X}_1^b$ is the leaping step. The extended ensemble $\widehat{\mathbf{X}}^b$ is given by

$$\widehat{\mathbf{X}}^b = \begin{bmatrix} X_0^b \\ X_1^b \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0^{b[1]}, \mathbf{x}_0^{b[2]}, \ldots, \mathbf{x}_0^{b[N]} \\ \mathbf{x}_1^{b[1]}, \mathbf{x}_1^{b[2]}, \ldots, \mathbf{x}_1^{b[N]} \end{bmatrix} \in \mathbb{R}^{2n\times N} \tag{4.6}$$

and the covariance matrix of $\widetilde{\mathbf{X}}$, $\mathbf{cov}(\widetilde{\mathbf{X}})$ is in the form of:

$$\begin{bmatrix} \mathbf{P}_0^b & \mathbf{\Delta X}_0 \cdot \mathbf{\Delta X}_1^T \\ \mathbf{\Delta X}_1 \cdot \mathbf{\Delta X}_0^T & \mathbf{P}_1^b \end{bmatrix}$$

But in real-life scenarios, we only have access to one observation $\{y_0^k\}_{k=0}^M \sim \mathcal{N}(\mathcal{H}(\mathbf{x}_k), \mathbf{R}_k)$, so the evolution of the system is depicted in figure 4.6

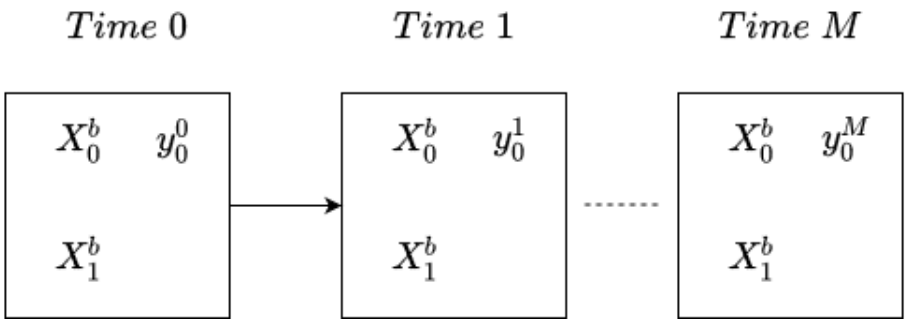


FIGURE 4.6: Time evolution of the extended ensemble

Our goal is to propagate the update over the intermediate step in the assimilation process, instead of updating both of them.

4.3 A data assimilation tool

The design, testing, and validation of data assimilation schemes are, most of the time, a time-consuming task with repetitive steps involved, which are replicated by each investigator in a science laboratory. Not only in the research field but in the academic field, we believe there is an opportunity in offering a package that will help to test and validate our models with a physical model already implemented and literature methods to compare.

4.3.1 AML-CS Package

The Python Package Applied Math and Computer Science Lab - Data Assimilation (**AMLCS-DA**) is a toolbox that allows you to develop, test, and use sequential data assimilation methods easily. The package is released with three well-known sequential data assimilation methods: the LETKF, the LEnKF, and the EnKF-MC, all of them detailed in section 2. Of course, you can add your methods as needed; this is detailed further. The general structure of our package can be seen in figure 4.7. The amlcs folder contains the necessary files (classes) to run (or use) our toolbox. The *models* folder defines the numerical model to be employed during forecast steps. The **AMLCS-DA** package is released with the SPEEDY model with five model resolutions. However, other resolutions (and models) can be coupled into the **AMLCS-DA** package as well. To run the SPEEDY model with another resolution, we need to provide a folder with the necessary files (i.e., boundary conditions, sea surface temperature, etc.); the reader can consult (Molteni, 2003a) to know more about this numerical model and how to set up additional model resolutions. Moreover, we have highly modified the SPEEDY model to support NetCDF files (Rew and Davis, 1990) as inputs and outputs (solutions).

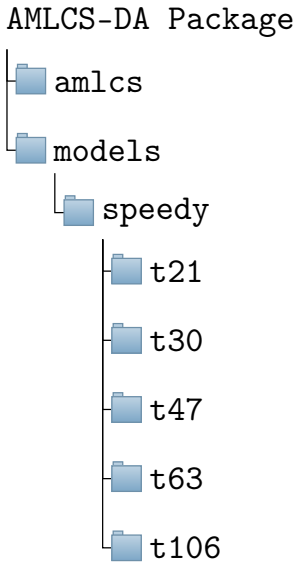


FIGURE 4.7: Folder structure of the **AMLCS-DA** package.

To start, the **AMLCS-DA** package can be employed in two different manners: as a final user to test (and to compare) sequential data assimilation methods (test suit) or as a developer to implement data assimilation methods

Such tool contains 3 main modules:

1. amlcs_pre.py: This file contains all the logic and routines related to model initialization and instancing, here are created:

   - Model's initial condition
   - Model's free run (with no DA)
   - Reference solution for each assimilation step
   - Ensemble members
   - Model files with their respective settings.

   This module receives a configuration file (a CSV) which contains:

   - *Nens*, defines the number of ensemble members to use in the assimilation process.
   - *M*, defines the time window in which the model will run.
   - *res_name*, chooses the model resolution (or specification) which the model can be used.
   - *per*, sets the perturbation value for the observations.
   - *obs_steps*, set how much times the observations will be taken. Can be hours, days, or any other model unit.
   - *ini_steps*, this referrers the number of days to use in the initial conditions construction.
   - *ini_times*, this referrers the number of times which the initial condition will be propagated.
   - *syn_tests*, set if synthetic observations will be used or not (a boolean).

- *folder_prep*, the path in which the initialization will be placed. Can be absolute or relative.
- *par* defines if the ensembles can be propagated in parallel during their construction.

This step is crucial, as the initial conditions and model setting will be used in the assimilation step. This module also creates a configuration file to be used in the assimilation process, containing the information regarding the noise of the observations and propagation settings.

Each different initialization requires a new file containing its settings. Required files can be seen in Figure 4.8

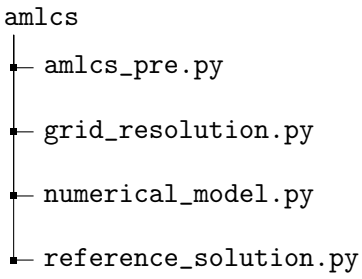


FIGURE 4.8: Tree of the amlcs folder. The necessary files to run (or use) our pre-processing toolbox.

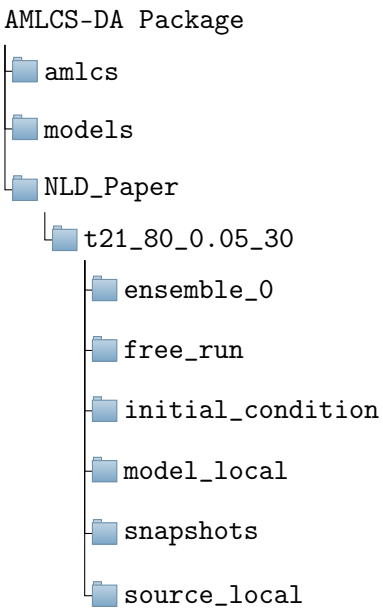


FIGURE 4.9: An example of our package tree once the pre-processing step has been performed.

2. amlcs_da.py: This module contains the Data assimilation process logic. The configuration file requires:

- *r*, the influence radius used in the localization process
- *s*, the sparsity of observations.

- *res_name*, the model resolution it has been compiled and setted previously.
- *method*, the name of the method, depending the available.
- *exp_settings*, the path in which the pre processing has been performed. Can be relative or absolute.
- *infla*, the inflation used in the method (if required)
- *err_obs*, the error in the observations for each variable. Used to build **R**.
- The list *obs_plc* denote which variables are observed in:

$$[u_0, v_0, T_0, q_0, \rho_0, u_1, v_1, T_1, q_1, \rho_1]\,,$$

  for instance, a list of the form: $[1,1,1,1,1,0,0,0,0,0]$ means all variables for the first step (in the leapfrog sense) are observed
- *list_snapshots*, a chosen list of moments in the assimilation process to be stored, saving the Analysis and the background for each.
- *option_mask*, which of the localization setting will be used, regarding the mentioned in the previous section.

This modules produces the assimilation results for a given configuration, creating:

- Model's free run (with no DA).
- Initial condition used.
- The errors for each variable and each level, in their respective CSV file.
- Reference solution used.
- The snapshots for each time in *list_snapshots*.

Once the experimental settings are built, we are ready to test data assimilation methods. As we mentioned before, three sequential methods are released in this package: LEnKF, LETKF, and EnKF-MC. The assimilation step relies on the Python classes defined in figure 4.10.

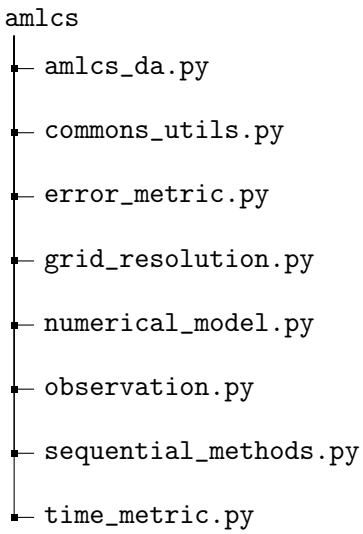


FIGURE 4.10: Tree of the amlcs folder. The necessary files to run (or use) our assimilation toolbox.

3. Once the assimilation process is completed, we can obtain some valuable visualizations of our method's performance by using the post-processing modules of AMLCS-DA. In figure 4.11, we show the Python files of our package to visualize (and analyze) the experimental results.

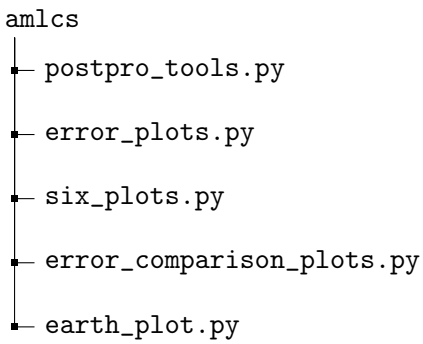


FIGURE 4.11: Tree of the amlcs folder. The necessary files to run (or use) our visualization toolbox.

The visualization module requires, as the previous ones, a configuration file that can contain multiple paths and plots specifications. This process will create the desired plots in a folder called "plots". This module consists of three sub-modules:

- a sub-module to compare results from multiple sequential data assimilation methods: we consider plots of the error evolution (analysis) and the error across numerical layers (analysis mean), respectively, we required a configuration (csv) file with the following parameters (headers):
  - mc_path: configuration file (path) of the EnKF-MC-Obs results,
  - letkf_path: configuration file (path) of the LETKF results,
  - lenkf_path: configuration file (path) of the LEnKF results,
  - resolution: the model resolution of experiments,
  - type: type of the plot,
  - variable: variables to be used (by default, it sets all model variables),
  - levels: numerical layers utilized in the error evolution plot (by default, it sets all model pressure levels)
- a sub-module to plot the error evolution of background and analysis for a single sequential data assimilation method; similar to the previous one, this requires only the path of a single experiment.
- a sub-module to plot heat maps comparing different configurations for a specific method (i.e., by varying inflation factors, localization radius lengths, and sparsity of observational networks); the configuration parameters are as follows:
  - setting: The configuration employed during the pre-processing step, consider as an example "t21_80_0.05_30":
  - method: The name of assimilation method, "EnKF_MC_obs"
  - infla: The inflation factor times 100.
  - mask: The employed mask.
  - variable: Variables to be used. If not provided, sets all model variables by default.

  - levels: Levels utilized in the error evolution plot. If not provided, sets all model pressure levels by default.

- The spatial propagation of errors can be studied via the earth plot, which compares different methods across numerical layers. Plots can be done for background and analysis errors, and even more, no data-assimilation errors can be plotted. The parameters of this configuration file read:
  - experiment_name
  - resolution
  - type
  - times
  - level
  - variable

  Similar to previous modules, this module comes with default values, and therefore, all parameters are optional but, *times*.

These modules, uses the classes and methods contained in:

- numerical_model.py: this class contains all methods related to the numerical model used, its settings and configurations. Here is where the ensemble members are propagated, initial conditions are built here as well, besides all variables behaves and specifications. If model requires an intermediate step, as transforming data or storing mid data, they are performed here. All the process of loading and saving the final data are done here as well. All data is saved using NetCDF4 format.

- grid_resolution.py: contains all methods related to spatial information and localization of the model.

- observations.py: is the one who created the observational operator $\mathbf{H}$ and synthetic observations.

- sequential_methods.py: this file contains a class which defines a factory for data assimilation methods. Defines shared aspects as the mapping from state vectors to ensembles, load background, inflation and so on. But each data assimilation technique has to define its own class and methods. We provide in the package with
  - EnKF-MC onto observations space (Our proposal)
  - LETKF
  - LEnKF

  User can add new methods by following the structure given by the factory and those three.

Even more, additional models can be added using the numerical_model and observations class structure. The simplicity on our package relies as it requires to fill a simple CSV format to run, with no further modification on code, and can be modified following a defined class structure.

Each experimental setting can be run independently, as they create a local model copy so they made the changes they require to produce forecasts. Once the process is completed, it will delete the ensembles and the model copies.

A further description related on model configuration will be provided on Chapter 5

In order to use the **AMLCS-DA** package, the following requirements must be satisfied:

- Python 3.X
- gfortran compiler
- netCDF4 and netCDF4-fortran (Can be a custom build or installed via libnetcdff-dev)

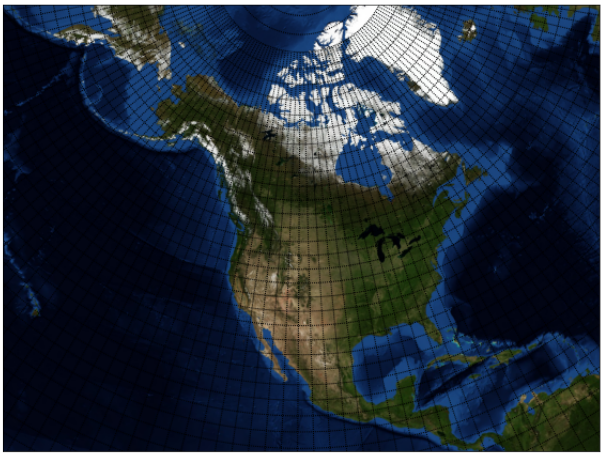
and we require the following Python packages:

- **netCDF4**
- **scikit-learn**
- **numpy**
- **pandas**
- **scipy**
- **matplotlib**
- **basemap**

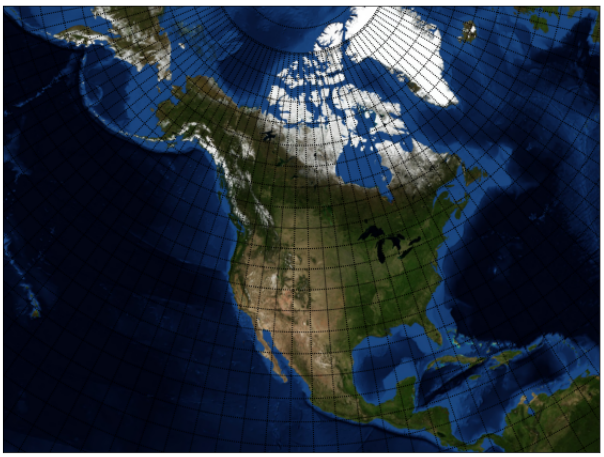
It is worth mentioning that the SPEEDY model requires a Linux system to run, but it can be employed on Windows, for instance, by using its Windows Subsystem for Linux (WSL).

4.3.2 Spatial Interpoler

As model configurations and observations can have different resolutions, we employ an spatial interpolator which takes into account their spatial resolution. As shown in 4.12, we can use the data provided by entities to produce datasets in the resolution of our model.



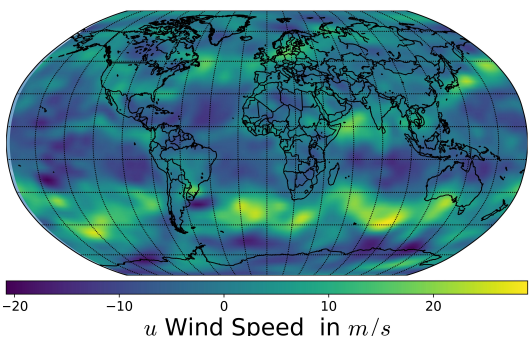
(A) NOAA ($144 \times 73 \times 17$)



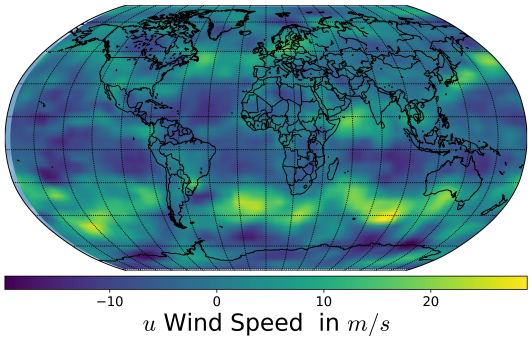
(B) SPEEDY $96 \times 48 \times 7$

FIGURE 4.12: Interpolation from NOAA dataset to SPEEDY grid resolution

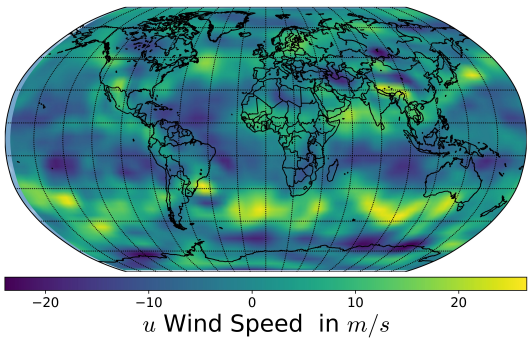
In particular, this interpolator converts units and transform variables for the NOAA dataset to SPEEDY specifications, as Relative Humidity and Specific Humidity. This transformation was made by using the procedure provided by National Center for Atmospheric Research (NCAR) using their language NCL (NCAR Command Language) which is an implementation of the formula provided by (Wallace and Hobbs, 2006).



(A) Real



(B) Interpolation



(C) Model Forecast

Chapter 5

Numerical Experiments

5.1 Numerical Model SPEEDY

The experiments are performed using the Simplified Parameterizations, privitivE-Equation DYnamics model (SPEEDY) (Molteni, 2003a; Bracco et al., 2004). The SPEEDY model is an Atmospheric General Circulation Model (AT-GCM) that mimics the behavior of the atmosphere across eight pressure levels (Miyoshi, 2011): $30mb$, $100mb$, $200mb$, $300mb$, $500mb$, $700mb$, $850mb$, and $925mb$. By default, this model employs a T-30 spectral resolution (96 zonal components and 48 meridional ones, across eight numerical layers) for the space discretization of numerical layers Molteni, 2003b; Kucharski, Molteni, and Bracco, 2006 but, this resolution can be modified, for example, to use: T-21 (32×64 grid components), T-47 (72×144), T-63 (96×192) and T-103 (160×320), each with 8 numerical layers. This model implements a two-steps leapfrog integration and the spatial grid components are shown in figure 5.1 for a single layer and different numerical resolutions. For all model resolutions, the physical variables are detailed in Table 5.1 with their corresponding units and the number of numerical layers.

Name	Notation	Units	Number of Layers
Temperature	T	K	8
Zonal Wind Component	u	m/s	8
Meridional Wind Component	v	m/s	8
Specific Humidity	Q	g/kg	8
Pressure	P	hPa	1

TABLE 5.1: Physical variables of the AT-GCM Speedy model.

(A) T-21 (B) T-30 (C) T-47

FIGURE 5.1: Model components for some spectral resolutions

The T-21 model resolution (32×64 grid components) is the one used for the horizontal space discretization in these experiments. Five model variables are part of the assimilation process: the temperature (K), the zonal and the meridional wind components (m/s), specific humidity (g/kg), and Surface Pressure (hPa). The total

number of model components is $n = 67,584$. The number of ensemble members is $N = 80$ for all the scenarios. The model state space is approximately 844 times larger than the number of ensemble members ($n \gg N$), which is very common in operational DA scenarios. Additional details of the experimental settings are described below, some of them are similar to those detailed in (Miyoshi, Kondo, and Imamura, 2014):

- Starting with a system in equilibrium, the model is integrated over a long time period to obtain an initial condition whose dynamics are consistent with those of the SPEEDY model.
- The initial condition is perturbed N times and propagated over a long-time period from which the initial background ensemble is obtained.
- We employ the trajectory of the initial condition as the reference one. This reference trajectory serves to build synthetic observations.
- We let the standard deviations of errors in the observations as follows:
  - Temperature 1 K.
  - Zonal Wind Component 1 m/s.
  - Meridional Wind Component 1 m/s.
  - Specific Humidity $10^{-3}\, g/kg$.
  - Pressure 100 hPa.
- The experiments are performed under perfect model assumptions.
- The number of assimilation steps reads $M = 30$.

Starting with the state of the system $\mathbf{x}^{\text{ref}}_{-3}$ at time t_{-3}, the model solution $\mathbf{x}^{\text{ref}}_{-3}$ is propagated in time over one year:

$$\mathbf{x}^{\text{ref}}_{-2} = \mathcal{M}_{t_{-3}\to t_{-2}}\left(\mathbf{x}^{\text{ref}}_{-3}\right).$$

The reference solution $\mathbf{x}^{\text{ref}}_{-2}$ is used to build a perturbed background solution:

$$\widehat{\mathbf{x}}^{\text{b}}_{-2} = \mathbf{x}^{\text{ref}}_{-2} + \boldsymbol{\varepsilon}^{\text{b}}_{-2}, \quad \boldsymbol{\varepsilon}^{\text{b}}_{-2} \sim \mathcal{N}\left(\mathbf{0}_n, \operatorname*{diag}_i\left\{\left(0.05\left\{\mathbf{x}^{\text{ref}}_{-2}\right\}_i\right)^2\right\}\right). \tag{5.1}$$

The perturbed background solution is propagated over another year to obtain the background solution at time t_{-1}:

$$\mathbf{x}^{\text{b}}_{-1} = \mathcal{M}_{t_{-2}\to t_{-1}}\left(\widehat{\mathbf{x}}^{\text{b}}_{-2}\right). \tag{5.2}$$

This model propagation attenuates the random noise introduced in (5.1) and makes the background state (5.2) consistent with the physics of the SPEEDY model. Then, the background state (5.2) is utilized in order to build an ensemble of perturbed background states:

$$\widehat{\mathbf{x}}^{\text{b}[i]}_{-1} = \mathbf{x}^{\text{b}}_{-1} + \boldsymbol{\varepsilon}^{\text{b}}_{-1}, \quad \boldsymbol{\varepsilon}^{\text{b}}_{-1} \sim \mathcal{N}\left(\mathbf{0}_n, \operatorname*{diag}_i\left\{\left(0.05\left\{\mathbf{x}^{\text{b}}_{-1}\right\}_i\right)^2\right\}\right), \quad 1 \le i \le N, \tag{5.3}$$

from which, after three months of model propagation, the initial ensemble is obtained at time t_0:

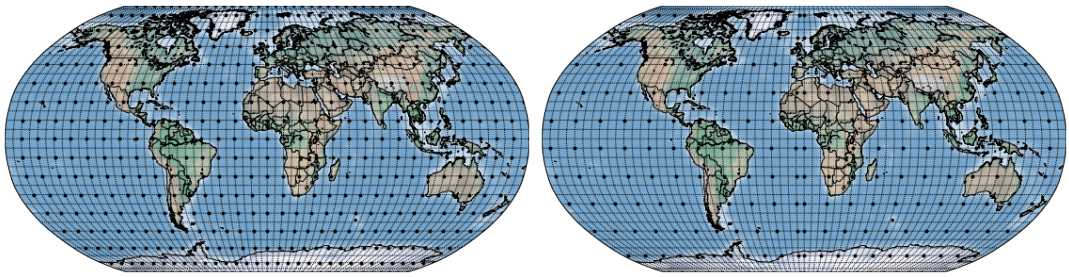
$$\mathbf{x}_0^{\mathrm{b}[i]} = \mathcal{M}_{t_{-1}\to t_0}\left(\widehat{\mathbf{x}}_{-1}^{\mathrm{b}[i]}\right).$$

Again, the model propagation of the perturbed ensemble ensures that the ensemble members are consistent with the physics of the numerical model.

The experiments are performed over a period of 24 days, where observations are taken every 2 days ($M = 12$). At time k synthetic observations are built as follows:

$$\mathbf{y}_k = \mathbf{H}_k \cdot \mathbf{x}_k^{\mathrm{ref}} + \boldsymbol{\varepsilon}_k, \quad \boldsymbol{\varepsilon}_k \sim \mathcal{N}\left(\mathbf{0_m}, \mathbf{R_k}\right), \quad \mathbf{R}_k = \mathrm{diag}_i\left\{\left(0.01\left\{\mathbf{H}_k\,\mathbf{x}_k^{\mathrm{ref}}\right\}_i\right)^2\right\}.$$

The observation operators $\mathbf{H}_k$ are fixed throughout the time interval. We perform experiments with several operators characterized by different proportions p of observed components from the model state $\mathbf{x}_k^{\mathrm{ref}}$ ($m \approx p \cdot n$). We consider four different values for p: 0.25, 0.11, 0.06 and 0.04 which represent 25%, 11 %, 6 % and 4 % of the total number of model components, respectively. Some of the observational networks used during the experiments are shown in Figure 5.2 with their corresponding percentage of observed components from the model state.



(A) 25% Observed (B) 11% Observed

FIGURE 5.2: Operational Observators

The analysis accuracy is measured by the root mean square error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\cdot\sum_{k=1}^{M}\left[\mathbf{x}_k^{\mathrm{ref}} - \mathbf{x}_k^{\mathrm{a}}\right]^T\cdot\left[\mathbf{x}_k^{\mathrm{ref}} - \mathbf{x}_k^{\mathrm{a}}\right]} \tag{5.4}$$

where $\mathbf{x}^{\mathrm{ref}} \in \mathbb{R}^{n\times 1}$ and $\mathbf{x}_k^{\mathrm{a}} \in \mathbb{R}^{n\times 1}$ are the reference and the analysis solutions at time k, respectively, and M is the number of assimilation times.

The different EnKF formulations were implemented using Python and specialized libraries such as scipy, numpy, and NetCDF.

The parameters used are as shown in Table 5.2

Parameters	Values
δ	3,5,7
p	25%,11%,6%,4%
α	1.02,1.04,1.06,1.08

TABLE 5.2: Parameters used for experiments

We consider sparse observational networks and vary the influence radius and inflation for previously mentioned formulations (EnKF-MC, LETKF, LEnKF), and for

each one, we performed assimilation taking into account the Leapfrog step (Leap) and assimilated using only the observed state (NoLeap). As we can observe in Table 5.3, which shows the log $RMSE$ for the zonal wind components, the error representativeness is degraded as we increase δ as there are no sufficient degrees of freedom to account for background error distribution. Additionally, we can notice that updates using the leaping step are statistically similar to the ones using only the observed model state, and updating. Data assimilation formulation produces better estimates than using only the model.

Tables 5.4, 5.5, 5.6, 5.7 show the RMSE for meridional wind component, specific humidity, temperature, and pressure, respectively.

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
1	100%	1.02	2.70	2.47	2.59	2.54	2.60	2.55	5.95
		1.04	2.72	2.51	2.59	2.54	2.60	2.55	
		1.06	2.74	2.54	2.59	2.54	2.60	2.55	
		1.08	2.77	2.58	2.59	2.54	2.60	2.55	
	25%	1.02	3.45	3.50	3.22	3.19	3.24	3.20	5.95
		1.04	3.52	3.62	3.22	3.19	3.24	3.20	
		1.06	3.71	3.83	3.22	3.19	3.24	3.20	
		1.08	3.80	4.07	3.22	3.19	3.24	3.20	
	11%	1.02	4.02	3.97	3.94	3.92	4.00	3.97	5.95
		1.04	4.09	4.05	3.94	3.92	4.00	3.97	
		1.06	4.21	4.37	3.94	3.92	4.00	3.97	
		1.08	4.42	4.62	3.94	3.92	4.00	3.97	
	6%	1.02	4.56	4.53	4.71	4.70	4.73	4.72	5.95
		1.04	4.62	4.61	4.71	4.70	4.73	4.72	
		1.06	4.74	4.79	4.71	4.70	4.73	4.72	
		1.08	4.62	5.04	4.71	4.70	4.73	4.72	
	4%	1.02	5.16	5.18	5.06	5.07	5.09	5.10	5.95
		1.04	5.24	5.19	5.06	5.07	5.09	5.10	
		1.06	5.34	5.29	5.06	5.07	5.09	5.10	
		1.08	5.38	5.36	5.06	5.07	5.09	5.10	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
3	100%	1.02	2.62	2.34	2.37	2.30	2.45	2.37	5.95
		1.04	2.63	2.36	2.37	2.30	2.45	2.37	
		1.06	2.64	2.39	2.37	2.30	2.45	2.37	
		1.08	2.67	2.44	2.37	2.30	2.45	2.37	
	25%	1.02	3.12	3.03	2.98	2.93	3.01	2.96	5.95
		1.04	3.14	3.05	2.98	2.93	3.01	2.96	
		1.06	3.17	3.09	2.98	2.93	3.01	2.96	
		1.08	3.24	3.17	2.98	2.93	3.01	2.96	
	11%	1.02	3.66	3.61	3.59	3.55	3.66	3.62	5.95
		1.04	3.68	3.63	3.59	3.55	3.66	3.62	
		1.06	3.72	3.66	3.59	3.55	3.66	3.62	
		1.08	3.78	3.72	3.59	3.55	3.66	3.62	
	6%	1.02	4.23	4.19	4.19	4.16	4.24	4.21	5.95
		1.04	4.25	4.21	4.19	4.16	4.24	4.21	
		1.06	4.29	4.25	4.19	4.16	4.24	4.21	
		1.08	4.35	4.32	4.19	4.16	4.24	4.21	
	4%	1.02	4.48	4.46	4.44	4.42	4.47	4.46	5.95
		1.04	4.53	4.50	4.44	4.42	4.47	4.46	
		1.06	4.59	4.56	4.44	4.42	4.47	4.46	
		1.08	4.70	4.66	4.44	4.42	4.47	4.46	
5	100%	1.02	2.76	2.40	2.32	2.13	2.44	2.34	5.95
		1.04	2.75	2.40	2.32	2.13	2.44	2.34	
		1.06	2.75	2.41	2.32	2.13	2.44	2.34	
		1.08	2.76	2.44	2.32	2.13	2.44	2.34	
	25%	1.02	3.21	3.12	2.90	2.84	2.97	2.91	5.95
		1.04	3.21	3.12	2.90	2.84	2.97	2.91	
		1.06	3.22	3.13	2.90	2.84	2.97	2.91	
		1.08	3.25	3.17	2.90	2.84	2.97	2.91	
	11%	1.02	3.75	3.66	3.49	3.45	3.57	3.52	5.95
		1.04	3.74	3.65	3.49	3.45	3.57	3.52	
		1.06	3.74	3.67	3.49	3.45	3.57	3.52	
		1.08	3.77	3.70	3.49	3.45	3.57	3.52	
	6%	1.02	4.24	4.20	4.10	4.06	4.16	4.13	5.95
		1.04	4.23	4.20	4.10	4.06	4.16	4.13	
		1.06	4.24	4.21	4.10	4.06	4.16	4.13	
		1.08	4.27	4.24	4.10	4.06	4.16	4.13	
	4%	1.02	4.45	4.20	4.30	4.28	4.36	4.34	5.95
		1.04	4.45	4.42	4.30	4.28	4.36	4.34	
		1.06	4.47	4.44	4.30	4.28	4.36	4.34	
		1.08	4.50	4.47	4.30	4.28	4.36	4.34	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
7	100%	1.02	2.92	2.52	2.34	2.25	2.51	2.42	5.95
		1.04	2.90	2.49	2.34	2.25	2.51	2.42	
		1.06	2.89	2.49	2.34	2.25	2.51	2.42	
		1.08	2.89	2.50	2.34	2.25	2.51	2.42	
	25%	1.02	3.34	3.23	2.90	2.83	3.02	2.95	5.95
		1.04	3.19	3.21	2.90	2.83	3.02	2.95	
		1.06	3.24	3.21	2.90	2.83	3.02	2.95	
		1.08	3.21	3.23	2.90	2.83	3.02	2.95	
	11%	1.02	3.81	3.73	3.50	3.46	3.60	3.54	5.95
		1.04	3.78	3.71	3.50	3.46	3.60	3.54	
		1.06	3.79	3.71	3.50	3.46	3.60	3.54	
		1.08	3.80	3.73	3.50	3.46	3.60	3.54	
	6%	1.02	4.28	4.28	4.07	4.03	4.15	4.14	5.95
		1.04	4.28	4.25	4.07	4.03	4.15	4.14	
		1.06	4.26	4.24	4.07	4.03	4.15	4.14	
		1.08	4.28	4.25	4.07	4.03	4.15	4.14	
	4%	1.02	4.52	4.53	4.27	4.25	4.34	4.31	5.95
		1.04	4.48	4.46	4.27	4.25	4.34	4.31	
		1.06	4.48	4.46	4.27	4.25	4.34	4.31	
		1.08	4.49	4.47	4.27	4.25	4.34	4.31	

TABLE 5.3: $\log(RMSE)$ for zonal wind components, using a sparse network

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
1	100%	1.02	2.65	2.47	2.57	2.51	2.59	2.53	5.90
		1.04	2.68	2.51	2.57	2.51	2.59	2.53	
		1.06	2.69	2.55	2.57	2.51	2.59	2.53	
		1.08	2.72	2.60	2.57	2.51	2.59	2.53	
	25%	1.02	3.32	3.22	3.19	3.15	3.20	3.16	5.90
		1.04	3.35	3.27	3.19	3.15	3.20	3.16	
		1.06	3.41	3.38	3.19	3.15	3.20	3.16	
		1.08	3.47	3.52	3.19	3.15	3.20	3.16	
	11%	1.02	3.85	3.74	3.88	3.85	3.94	3.90	5.90
		1.04	3.89	3.79	3.88	3.85	3.94	3.90	
		1.06	3.98	3.93	3.88	3.85	3.94	3.90	
		1.08	4.07	4.05	3.88	3.85	3.94	3.90	
	6%	1.02	4.42	4.39	4.69	4.68	4.71	4.70	5.90
		1.04	4.48	4.47	4.69	4.68	4.71	4.70	
		1.06	4.56	4.58	4.69	4.68	4.71	4.70	
		1.08	4.45	4.74	4.69	4.68	4.71	4.70	
	4%	1.02	5.03	5.01	5.08	5.08	5.10	5.09	5.90
		1.04	5.08	5.06	5.08	5.08	5.10	5.09	
		1.06	5.16	5.13	5.08	5.08	5.10	5.09	
		1.08	5.17	5.17	5.08	5.08	5.10	5.09	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
3	100%	1.02	2.52	2.33	2.36	2.29	2.45	2.37	5.90
		1.04	2.53	2.35	2.36	2.29	2.45	2.37	
		1.06	2.54	2.38	2.36	2.29	2.45	2.37	
		1.08	2.57	2.43	2.36	2.29	2.45	2.37	
	25%	1.02	3.05	2.97	2.96	2.90	2.99	2.93	5.90
		1.04	3.07	2.99	2.96	2.90	2.99	2.93	
		1.06	3.09	3.01	2.96	2.90	2.99	2.93	
		1.08	3.12	3.05	2.96	2.90	2.99	2.93	
	11%	1.02	3.50	3.45	3.50	3.45	3.54	3.50	5.90
		1.04	3.52	3.46	3.50	3.45	3.54	3.50	
		1.06	3.53	3.48	3.50	3.45	3.54	3.50	
		1.08	3.56	3.51	3.50	3.45	3.54	3.50	
	6%	1.02	4.14	4.09	4.09	4.05	4.15	4.11	5.90
		1.04	4.16	4.11	4.09	4.05	4.15	4.11	
		1.06	4.19	4.15	4.09	4.05	4.15	4.11	
		1.08	4.25	4.20	4.09	4.05	4.15	4.11	
	4%	1.02	4.53	4.49	4.47	4.46	4.50	4.48	5.90
		1.04	4.57	4.53	4.47	4.46	4.50	4.48	
		1.06	4.63	4.59	4.47	4.46	4.50	4.48	
		1.08	4.71	4.68	4.47	4.46	4.50	4.48	
5	100%	1.02	2.74	2.40	2.31	2.12	2.43	2.33	5.90
		1.04	2.73	2.40	2.31	2.12	2.43	2.33	
		1.06	2.73	2.41	2.31	2.12	2.43	2.33	
		1.08	2.73	2.44	2.31	2.12	2.43	2.33	
	25%	1.02	3.19	3.09	2.87	2.79	2.93	2.87	5.90
		1.04	3.18	3.08	2.87	2.79	2.93	2.87	
		1.06	3.18	3.08	2.87	2.79	2.93	2.87	
		1.08	3.20	3.10	2.87	2.79	2.93	2.87	
	11%	1.02	3.62	3.53	3.35	3.30	3.42	3.37	5.90
		1.04	3.60	3.52	3.35	3.30	3.42	3.37	
		1.06	3.59	3.51	3.35	3.30	3.42	3.37	
		1.08	3.60	3.52	3.35	3.30	3.42	3.37	
	6%	1.02	4.16	4.11	4.00	3.96	4.06	4.03	5.90
		1.04	4.13	4.10	4.00	3.96	4.06	4.03	
		1.06	4.14	4.10	4.00	3.96	4.06	4.03	
		1.08	4.16	4.13	4.00	3.96	4.06	4.03	
	4%	1.02	4.48	4.21	4.35	4.33	4.40	4.39	5.90
		1.04	4.48	4.44	4.35	4.33	4.40	4.39	
		1.06	4.49	4.46	4.35	4.33	4.40	4.39	
		1.08	4.52	4.49	4.35	4.33	4.40	4.39	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
7	100%	1.02	2.89	2.52	2.33	2.23	2.49	2.41	5.90
		1.04	2.87	2.49	2.33	2.23	2.49	2.41	
		1.06	2.86	2.49	2.33	2.23	2.49	2.41	
		1.08	2.85	2.49	2.33	2.23	2.49	2.41	
	25%	1.02	3.31	3.20	2.85	2.77	2.98	2.91	5.90
		1.04	3.14	3.18	2.85	2.77	2.98	2.91	
		1.06	3.19	3.16	2.85	2.77	2.98	2.91	
		1.08	3.17	3.17	2.85	2.77	2.98	2.91	
	11%	1.02	3.69	3.62	3.33	3.28	3.43	3.37	5.90
		1.04	3.65	3.59	3.33	3.28	3.43	3.37	
		1.06	3.64	3.57	3.33	3.28	3.43	3.37	
		1.08	3.64	3.57	3.33	3.28	3.43	3.37	
	6%	1.02	4.20	4.20	3.96	3.91	4.05	4.02	5.90
		1.04	4.19	4.16	3.96	3.91	4.05	4.02	
		1.06	4.17	4.15	3.96	3.91	4.05	4.02	
		1.08	4.18	4.15	3.96	3.91	4.05	4.02	
	4%	1.02	4.55	4.56	4.31	4.30	4.38	4.36	5.90
		1.04	4.50	4.48	4.31	4.30	4.38	4.36	
		1.06	4.50	4.47	4.31	4.30	4.38	4.36	
		1.08	4.50	4.48	4.31	4.30	4.38	4.36	

TABLE 5.4: $\log(RMSE)$ for meridional wind components, using a sparse network

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
1	100%	1.02	2.26	2.16	2.27	2.23	2.28	2.23	5.25
		1.04	2.29	2.20	2.27	2.23	2.28	2.23	
		1.06	2.33	2.25	2.27	2.23	2.28	2.23	
		1.08	2.38	2.30	2.27	2.23	2.28	2.23	
	25%	1.02	2.98	2.93	2.84	2.81	2.84	2.80	5.25
		1.04	3.03	3.01	2.84	2.81	2.84	2.80	
		1.06	3.12	3.12	2.84	2.81	2.84	2.80	
		1.08	3.21	3.26	2.84	2.81	2.84	2.80	
	11%	1.02	3.46	3.40	3.40	3.38	3.43	3.40	5.25
		1.04	3.53	3.49	3.40	3.38	3.43	3.40	
		1.06	3.62	3.64	3.40	3.38	3.43	3.40	
		1.08	3.75	3.79	3.40	3.38	3.43	3.40	
	6%	1.02	3.96	3.95	4.17	4.15	4.18	4.16	5.25
		1.04	4.04	4.05	4.17	4.15	4.18	4.16	
		1.06	4.21	4.22	4.17	4.15	4.18	4.16	
		1.08	4.11	4.48	4.17	4.15	4.18	4.16	
	4%	1.02	4.37	4.38	4.43	4.44	4.45	4.46	5.25
		1.04	4.47	4.46	4.43	4.44	4.45	4.46	
		1.06	4.68	4.66	4.43	4.44	4.45	4.46	
		1.08	4.87	4.91	4.43	4.44	4.45	4.46	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
3	100%	1.02	2.22	2.07	2.10	2.05	2.19	2.12	5.25
		1.04	2.23	2.09	2.10	2.05	2.19	2.12	
		1.06	2.25	2.12	2.10	2.05	2.19	2.12	
		1.08	2.29	2.16	2.10	2.05	2.19	2.12	
	25%	1.02	2.78	2.71	2.70	2.66	2.73	2.68	5.25
		1.04	2.80	2.73	2.70	2.66	2.73	2.68	
		1.06	2.84	2.76	2.70	2.66	2.73	2.68	
		1.08	2.88	2.81	2.70	2.66	2.73	2.68	
	11%	1.02	3.24	3.19	3.13	3.09	3.17	3.13	5.25
		1.04	3.27	3.21	3.13	3.09	3.17	3.13	
		1.06	3.31	3.25	3.13	3.09	3.17	3.13	
		1.08	3.36	3.30	3.13	3.09	3.17	3.13	
	6%	1.02	3.72	3.67	3.62	3.59	3.67	3.64	5.25
		1.04	3.77	3.71	3.62	3.59	3.67	3.64	
		1.06	3.82	3.77	3.62	3.59	3.67	3.64	
		1.08	3.90	3.85	3.62	3.59	3.67	3.64	
	4%	1.02	3.96	3.92	3.87	3.87	3.88	3.87	5.25
		1.04	4.02	3.98	3.87	3.87	3.88	3.87	
		1.06	4.12	4.08	3.87	3.87	3.88	3.87	
		1.08	4.28	4.23	3.87	3.87	3.88	3.87	
5	100%	1.02	2.36	2.13	2.06	1.93	2.23	2.13	5.25
		1.04	2.35	2.13	2.06	1.93	2.23	2.13	
		1.06	2.36	2.14	2.06	1.93	2.23	2.13	
		1.08	2.36	2.16	2.06	1.93	2.23	2.13	
	25%	1.02	2.84	2.76	2.67	2.62	2.75	2.69	5.25
		1.04	2.84	2.76	2.67	2.62	2.75	2.69	
		1.06	2.85	2.77	2.67	2.62	2.75	2.69	
		1.08	2.86	2.78	2.67	2.62	2.75	2.69	
	11%	1.02	3.26	3.19	3.07	3.03	3.13	3.09	5.25
		1.04	3.25	3.19	3.07	3.03	3.13	3.09	
		1.06	3.25	3.20	3.07	3.03	3.13	3.09	
		1.08	3.27	3.21	3.07	3.03	3.13	3.09	
	6%	1.02	3.68	3.66	3.56	3.53	3.61	3.58	5.25
		1.04	3.68	3.67	3.56	3.53	3.61	3.58	
		1.06	3.70	3.68	3.56	3.53	3.61	3.58	
		1.08	3.73	3.72	3.56	3.53	3.61	3.58	
	4%	1.02	3.92	3.62	3.77	3.76	3.81	3.80	5.25
		1.04	3.93	3.90	3.77	3.76	3.81	3.80	
		1.06	3.96	3.93	3.77	3.76	3.81	3.80	
		1.08	4.00	3.97	3.77	3.76	3.81	3.80	

Continued on next page

δ	p	α	EnKF-MC		LETKF		LEnK		NODA
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
7	100%	1.02	2.50	2.22	2.08	2.01	2.31	2.23	5.25
		1.04	2.48	2.21	2.08	2.01	2.31	2.23	
		1.06	2.47	2.20	2.08	2.01	2.31	2.23	
		1.08	2.46	2.21	2.08	2.01	2.31	2.23	
	25%	1.02	2.94	2.85	2.68	2.63	2.81	2.76	5.25
		1.04	2.79	2.82	2.68	2.63	2.81	2.76	
		1.06	2.84	2.82	2.68	2.63	2.81	2.76	
		1.08	2.82	2.82	2.68	2.63	2.81	2.76	
	11%	1.02	3.32	3.25	3.08	3.04	3.17	3.12	5.25
		1.04	3.30	3.23	3.08	3.04	3.17	3.12	
		1.06	3.29	3.23	3.08	3.04	3.17	3.12	
		1.08	3.30	3.23	3.08	3.04	3.17	3.12	
	6%	1.02	3.75	3.75	3.54	3.50	3.62	3.60	5.25
		1.04	3.75	3.73	3.54	3.50	3.62	3.60	
		1.06	3.74	3.73	3.54	3.50	3.62	3.60	
		1.08	3.75	3.74	3.54	3.50	3.62	3.60	
	4%	1.02	3.99	4.00	3.75	3.74	3.80	3.78	5.25
		1.04	3.96	3.94	3.75	3.74	3.80	3.78	
		1.06	3.97	3.94	3.75	3.74	3.80	3.78	
		1.08	3.99	3.96	3.75	3.74	3.80	3.78	

TABLE 5.5: $\log(RMSE)$ for temperature, using a sparse network

δ	p	α	EnKF-MC		LETKF		LEnK		NODA
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
1	100%	1.02	0.49	-3.32	0.23	-3.41	1.89	1.88	3.65
		1.04	0.49	-3.32	0.23	-3.41	1.89	1.88	
		1.06	0.51	-3.31	0.23	-3.41	1.89	1.88	
		1.08	0.52	-3.31	0.23	-3.41	1.89	1.88	
	25%	1.02	1.01	0.90	0.78	0.59	2.04	2.03	3.65
		1.04	1.04	0.93	0.78	0.59	2.04	2.03	
		1.06	1.08	0.99	0.78	0.59	2.04	2.03	
		1.08	1.12	1.05	0.78	0.59	2.04	2.03	
	11%	1.02	1.77	1.74	1.85	1.82	2.36	2.33	3.65
		1.04	1.82	1.80	1.85	1.82	2.36	2.33	
		1.06	1.90	1.90	1.85	1.82	2.36	2.33	
		1.08	1.99	2.00	1.85	1.82	2.36	2.33	
	6%	1.02	2.43	2.40	2.61	2.59	2.80	2.79	3.65
		1.04	2.53	2.50	2.61	2.59	2.80	2.79	
		1.06	2.65	2.63	2.61	2.59	2.80	2.79	
		1.08	2.53	2.77	2.61	2.59	2.80	2.79	
	4%	1.02	3.02	2.97	2.96	2.95	3.09	3.08	3.65
		1.04	3.09	3.05	2.96	2.95	3.09	3.08	
		1.06	3.18	3.14	2.96	2.95	3.09	3.08	
		1.08	3.21	3.20	2.96	2.95	3.09	3.08	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
3	100%	1.02	0.46	-3.31	0.39	-3.64	1.89	1.88	3.65
		1.04	0.46	-3.31	0.39	-3.64	1.89	1.88	
		1.06	0.46	-3.30	0.39	-3.64	1.89	1.88	
		1.08	0.47	-3.30	0.39	-3.64	1.89	1.88	
	25%	1.02	0.76	0.55	0.54	0.18	1.98	1.97	3.65
		1.04	0.76	0.56	0.54	0.18	1.98	1.97	
		1.06	0.77	0.57	0.54	0.18	1.98	1.97	
		1.08	0.79	0.59	0.54	0.18	1.98	1.97	
	11%	1.02	1.47	1.41	1.45	1.39	2.20	2.18	3.65
		1.04	1.48	1.43	1.45	1.39	2.20	2.18	
		1.06	1.51	1.45	1.45	1.39	2.20	2.18	
		1.08	1.54	1.48	1.45	1.39	2.20	2.18	
	6%	1.02	2.17	2.12	2.13	2.10	2.52	2.49	3.65
		1.04	2.20	2.15	2.13	2.10	2.52	2.49	
		1.06	2.24	2.19	2.13	2.10	2.52	2.49	
		1.08	2.30	2.26	2.13	2.10	2.52	2.49	
	4%	1.02	2.52	2.48	2.51	2.51	2.71	2.70	3.65
		1.04	2.59	2.56	2.51	2.51	2.71	2.70	
		1.06	2.69	2.66	2.51	2.51	2.71	2.70	
		1.08	2.83	2.79	2.51	2.51	2.71	2.70	
5	100%	1.02	0.63	-3.30	1.11	-3.31	1.89	1.88	3.65
		1.04	0.63	-3.30	1.11	-3.31	1.89	1.88	
		1.06	0.61	-3.29	1.11	-3.31	1.89	1.88	
		1.08	0.60	-3.29	1.11	-3.31	1.89	1.88	
	25%	1.02	0.88	0.68	0.59	0.19	1.97	1.96	3.65
		1.04	0.88	0.68	0.59	0.19	1.97	1.96	
		1.06	0.88	0.68	0.59	0.19	1.97	1.96	
		1.08	0.87	0.69	0.59	0.19	1.97	1.96	
	11%	1.02	1.53	1.44	1.38	1.32	2.14	2.11	3.65
		1.04	1.52	1.44	1.38	1.32	2.14	2.11	
		1.06	1.52	1.45	1.38	1.32	2.14	2.11	
		1.08	1.53	1.46	1.38	1.32	2.14	2.11	
	6%	1.02	2.08	2.05	2.02	1.99	2.45	2.43	3.65
		1.04	2.07	2.05	2.02	1.99	2.45	2.43	
		1.06	2.08	2.06	2.02	1.99	2.45	2.43	
		1.08	2.11	2.09	2.02	1.99	2.45	2.43	
	4%	1.02	2.47	2.21	2.36	2.34	2.62	2.62	3.65
		1.04	2.47	2.43	2.36	2.34	2.62	2.62	
		1.06	2.50	2.47	2.36	2.34	2.62	2.62	
		1.08	2.54	2.52	2.36	2.34	2.62	2.62	

Continued on next page

δ	p	α	EnKF-MC		LETKF		LEnK		NODA
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
7	100%	1.02	0.83	-3.28	1.10	-0.94	1.90	1.90	3.65
		1.04	0.80	-3.28	1.10	-0.94	1.90	1.90	
		1.06	0.78	-3.28	1.10	-0.94	1.90	1.90	
		1.08	0.77	-3.28	1.10	-0.94	1.90	1.90	
	25%	1.02	1.06	0.85	1.00	0.62	1.99	1.98	3.65
		1.04	0.92	0.85	1.00	0.62	1.99	1.98	
		1.06	0.97	0.85	1.00	0.62	1.99	1.98	
		1.08	0.96	0.85	1.00	0.62	1.99	1.98	
	11%	1.02	1.61	1.52	1.42	1.35	2.15	2.12	3.65
		1.04	1.59	1.51	1.42	1.35	2.15	2.12	
		1.06	1.58	1.51	1.42	1.35	2.15	2.12	
		1.08	1.59	1.51	1.42	1.35	2.15	2.12	
	6%	1.02	2.15	2.13	1.97	1.93	2.45	2.44	3.65
		1.04	2.15	2.11	1.97	1.93	2.45	2.44	
		1.06	2.14	2.10	1.97	1.93	2.45	2.44	
		1.08	2.15	2.11	1.97	1.93	2.45	2.44	
	4%	1.02	2.50	2.50	2.32	2.30	2.60	2.59	3.65
		1.04	2.47	2.46	2.32	2.30	2.60	2.59	
		1.06	2.49	2.46	2.32	2.30	2.60	2.59	
		1.08	2.51	2.48	2.32	2.30	2.60	2.59	

TABLE 5.6: $\log(RMSE)$ for specific humidity, using a sparse network

δ	p	α	EnKF-MC		LETKF		LEnK		NODA
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
1	100%	1.02	5.60	5.60	5.60	5.60	5.60	5.60	5.62
		1.04	5.60	5.60	5.60	5.60	5.60	5.60	
		1.06	5.60	5.60	5.60	5.60	5.60	5.60	
		1.08	5.60	5.61	5.60	5.60	5.60	5.60	
	25%	1.02	5.61	5.61	5.60	5.60	5.60	5.60	5.62
		1.04	5.61	5.61	5.60	5.60	5.60	5.60	
		1.06	5.61	5.61	5.60	5.60	5.60	5.60	
		1.08	5.61	5.61	5.60	5.60	5.60	5.60	
	11%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	6%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.43	5.61	5.61	5.61	5.61	5.61	
	4%	1.02	5.60	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.56	5.58	5.61	5.61	5.61	5.61	
	100%	1.02	5.60	5.60	5.60	5.60	5.60	5.60	5.62
		1.04	5.60	5.60	5.60	5.60	5.60	5.60	
		1.06	5.60	5.60	5.60	5.60	5.60	5.60	

Continued on next page

δ	p	α	**EnKF-MC**		**LETKF**		**LEnK**		**NODA**
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
		1.08	5.60	5.60	5.60	5.60	5.60	5.60	
	25%	1.02	5.60	5.60	5.60	5.60	5.60	5.61	5.62
		1.04	5.60	5.61	5.60	5.60	5.60	5.61	
		1.06	5.61	5.61	5.60	5.60	5.60	5.61	
		1.08	5.61	5.61	5.60	5.60	5.60	5.61	
	11%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	6%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	4%	1.02	5.61	5.60	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
5	100%	1.02	5.60	5.60	5.60	5.45	5.60	5.60	5.62
		1.04	5.60	5.60	5.60	5.45	5.60	5.60	
		1.06	5.60	5.60	5.60	5.45	5.60	5.60	
		1.08	5.60	5.60	5.60	5.45	5.60	5.60	
	25%	1.02	5.61	5.57	5.60	5.60	5.60	5.60	5.62
		1.04	5.61	5.61	5.60	5.60	5.60	5.60	
		1.06	5.61	5.61	5.60	5.60	5.60	5.60	
		1.08	5.61	5.61	5.60	5.60	5.60	5.60	
	11%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	6%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	4%	1.02	5.61	4.96	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	

Continued on next page

δ	p	α	EnKF-MC		LETKF		LEnK		NODA
			NoLeap	Leap	NoLeap	Leap	NoLeap	Leap	
7	100%	1.02	5.60	5.60	5.60	5.60	5.60	5.60	5.62
		1.04	5.60	5.60	5.60	5.60	5.60	5.60	
		1.06	5.60	5.60	5.60	5.60	5.60	5.60	
		1.08	5.60	5.60	5.60	5.60	5.60	5.60	
	25%	1.02	5.60	5.60	5.60	5.60	5.60	5.60	5.62
		1.04	5.15	5.60	5.60	5.60	5.60	5.60	
		1.06	5.38	5.61	5.60	5.60	5.60	5.60	
		1.08	5.33	5.61	5.60	5.60	5.60	5.60	
	11%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	6%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	
	4%	1.02	5.61	5.61	5.61	5.61	5.61	5.61	5.62
		1.04	5.61	5.61	5.61	5.61	5.61	5.61	
		1.06	5.61	5.61	5.61	5.61	5.61	5.61	
		1.08	5.61	5.61	5.61	5.61	5.61	5.61	

TABLE 5.7: $\log(RMSE)$ for surface pressure, using a sparse network

The Figures 5.3 to 5.102 shows the evolution thought the assimilation time window for all the variables, comparing them using the leaping step and without using it. Clearly, when the radius of influence is increased the analysis corrections are impacted by spurious correlations. Yet, we can observe the assimilation is not greatly impacted if we do not have access to the leaping step, and instead update it using only the model state. When data errors components are uncorrelated δ can be seen as a free parameter and the choice can be based on the "optimal performance of the filter".

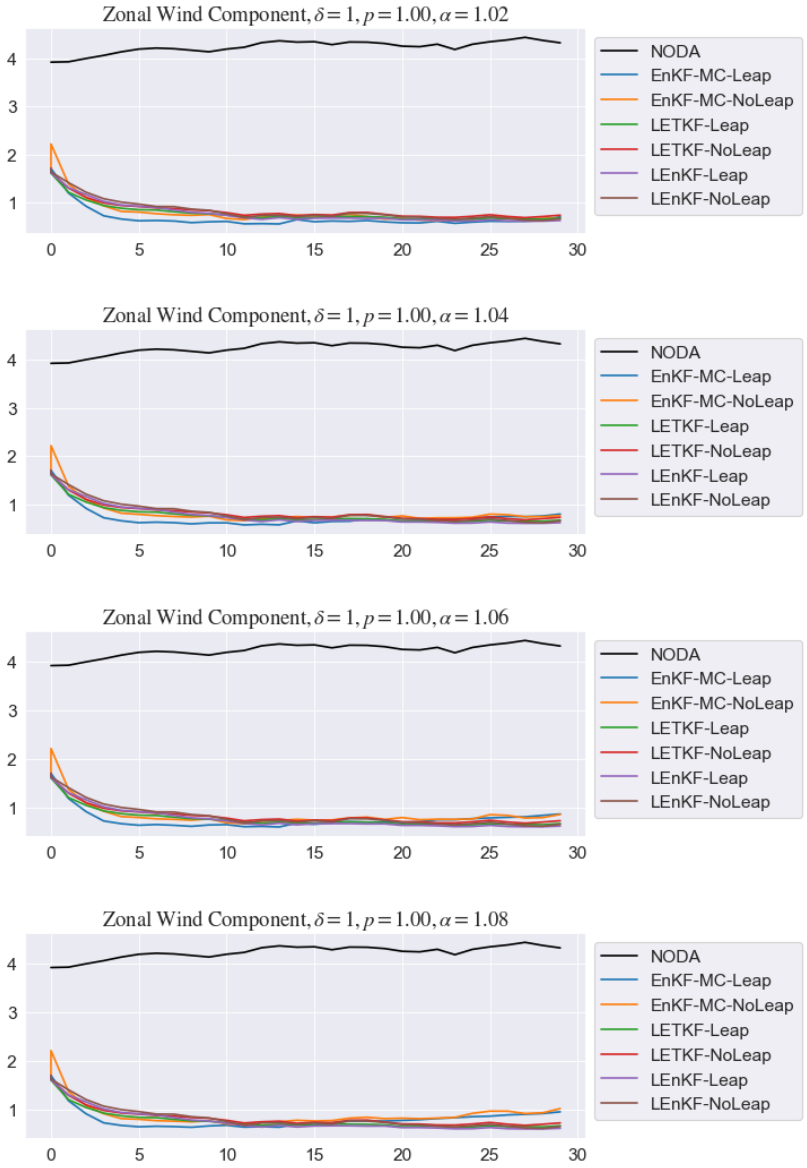


FIGURE 5.3: Time evolution of Zonal Wind Components for $\delta = 1$ and $p = 100\%$ varying α

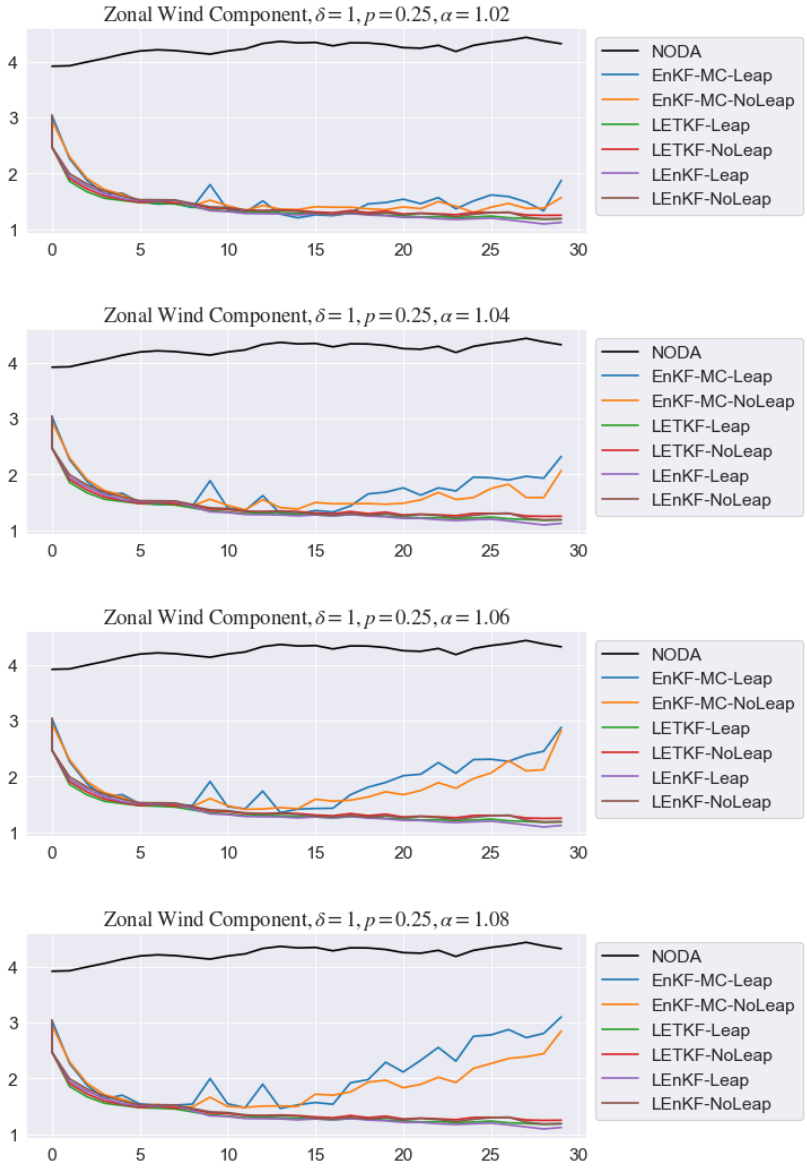


FIGURE 5.4: Time evolution of Zonal Wind Components for $\delta = 1$ and $p = 25\%$ varying α

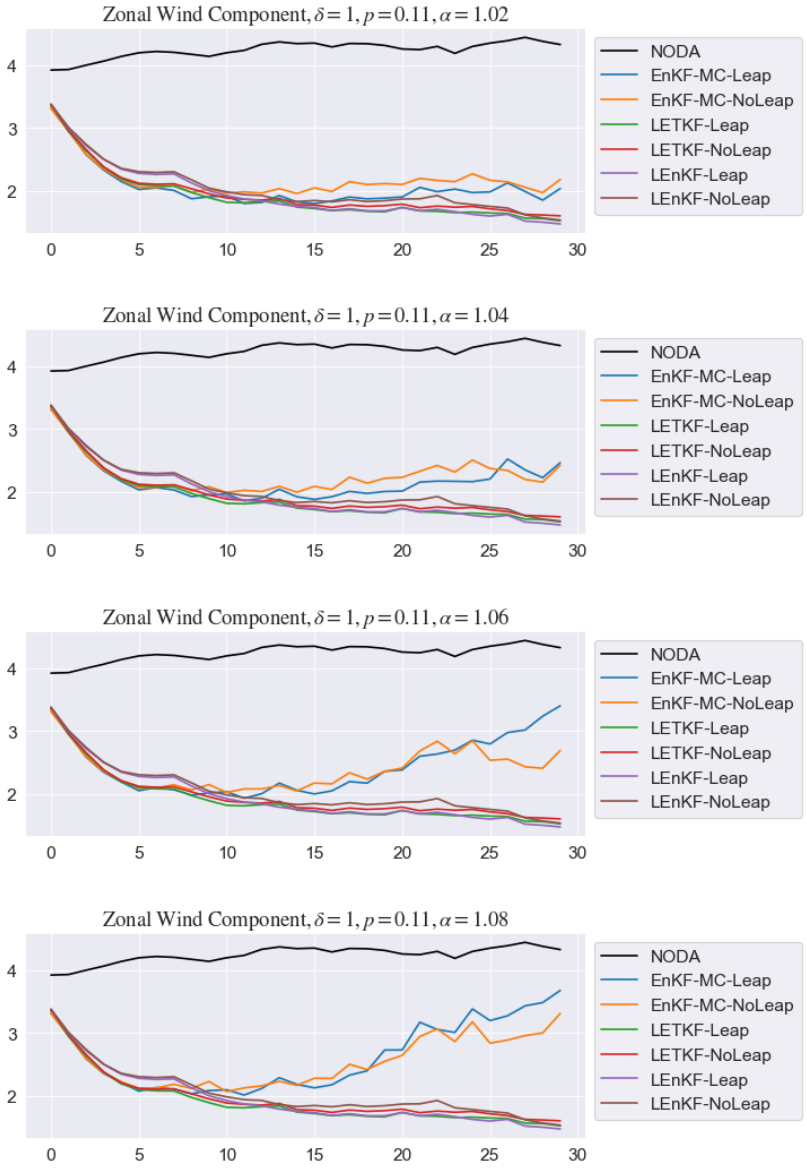


FIGURE 5.5: Time evolution of Zonal Wind Components for $\delta = 1$ and $p = 11\%$ varying α

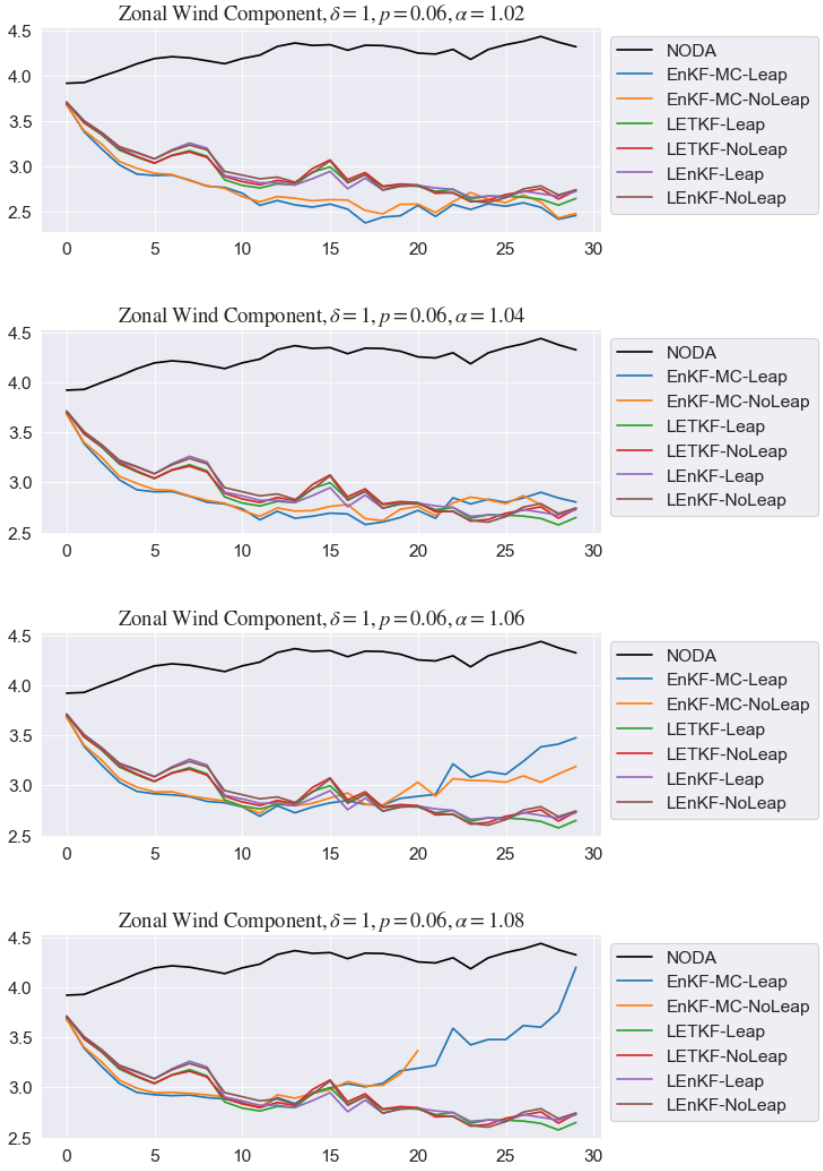


FIGURE 5.6: Time evolution of Zonal Wind Components for $\delta = 1$ and $p = 6\%$ varying α

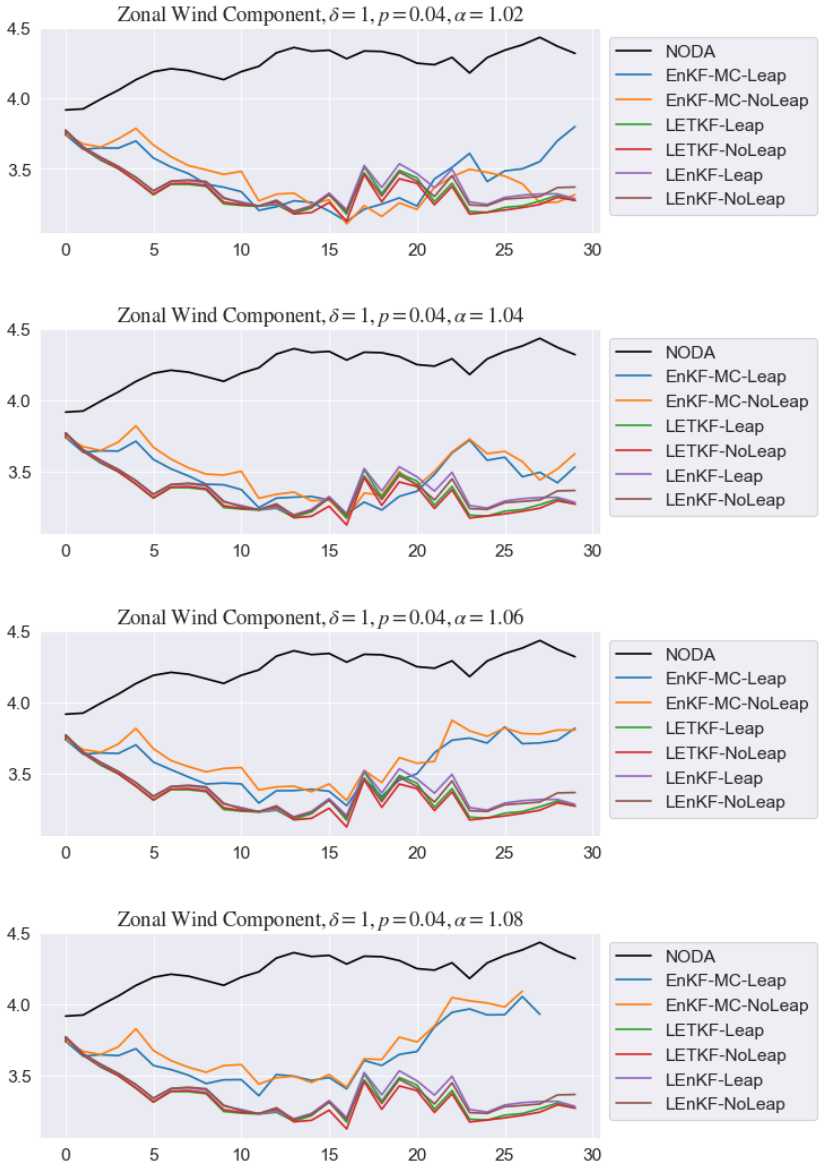


FIGURE 5.7: Time evolution of Zonal Wind Components for $\delta = 1$ and $p = 4\%$ varying α

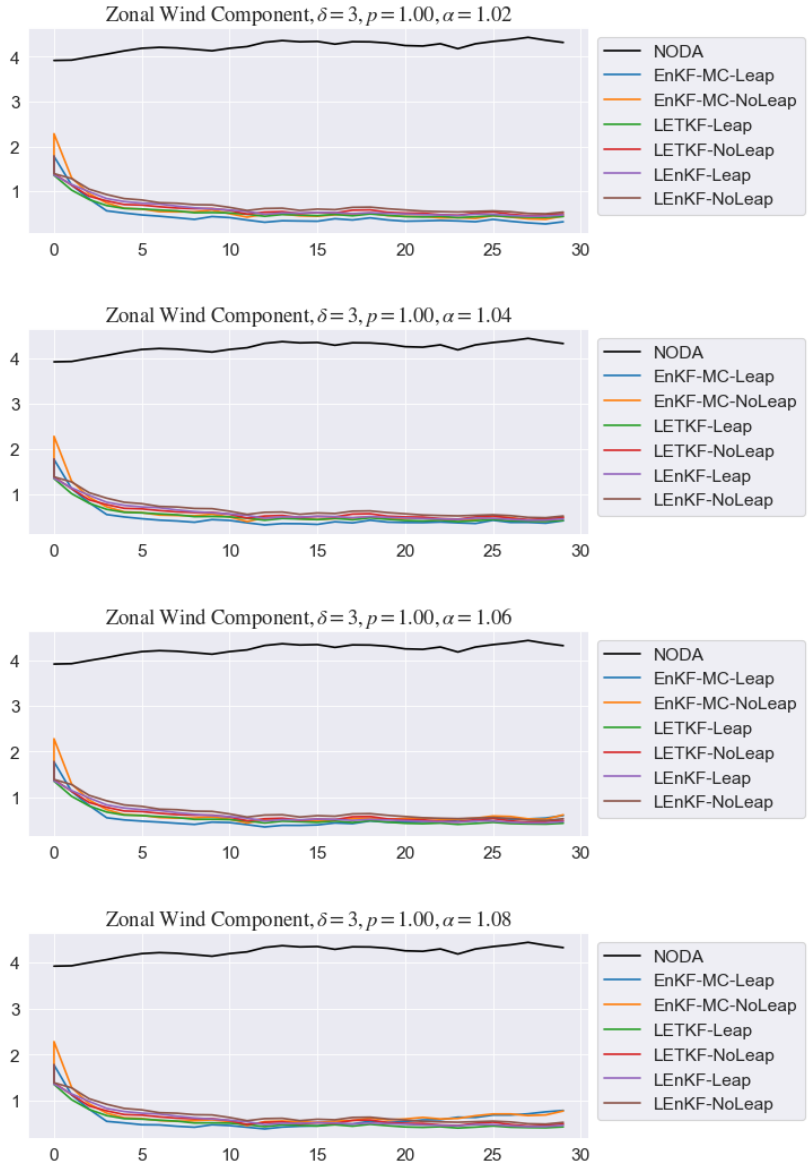


FIGURE 5.8: Time evolution of Zonal Wind Components for $\delta = 3$ and $p = 100\%$ varying α

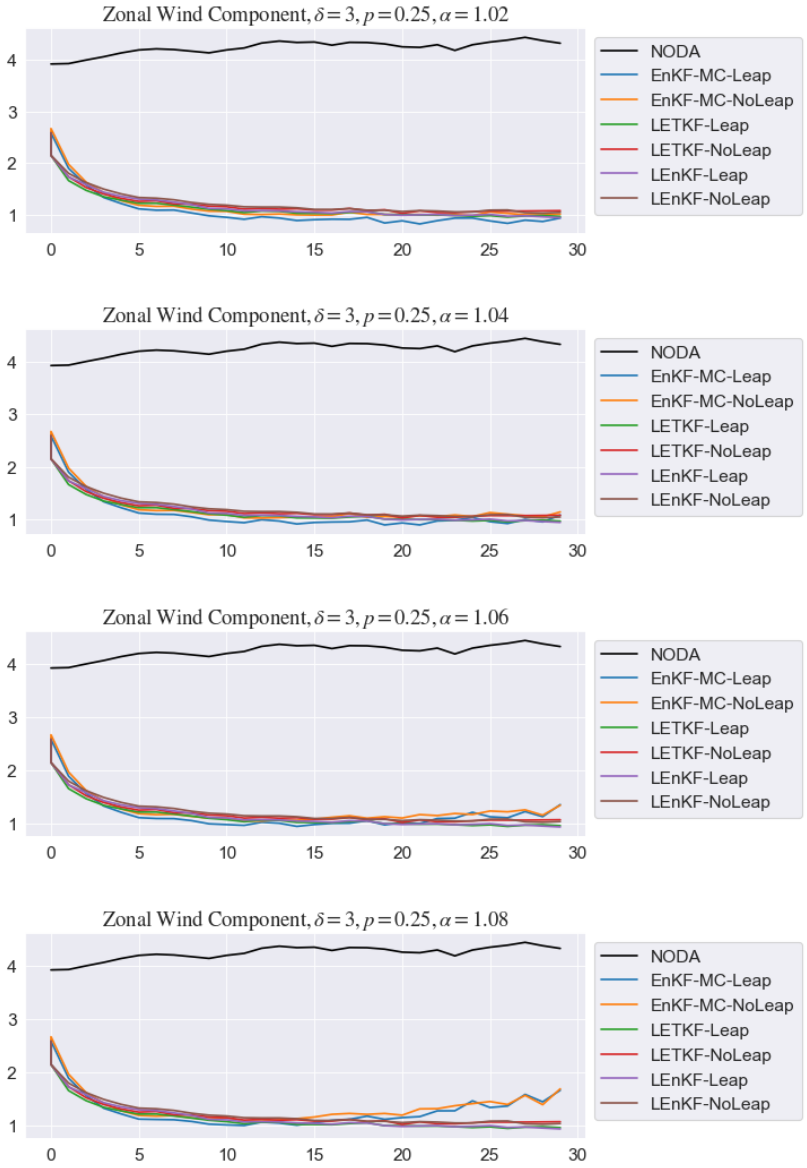


FIGURE 5.9: Time evolution of Zonal Wind Components for $\delta = 3$ and $p = 25\%$ varying α

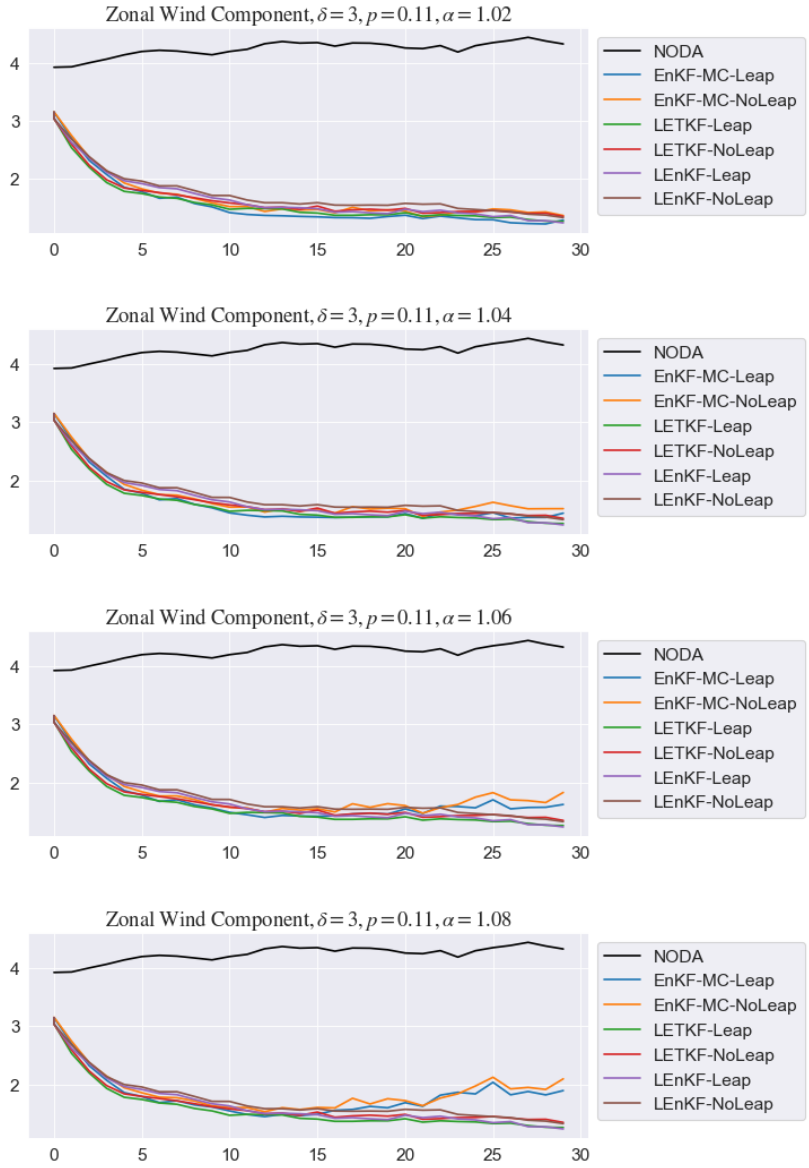


FIGURE 5.10: Time evolution of Zonal Wind Components for $\delta = 3$ and $p = 11\%$ varying α

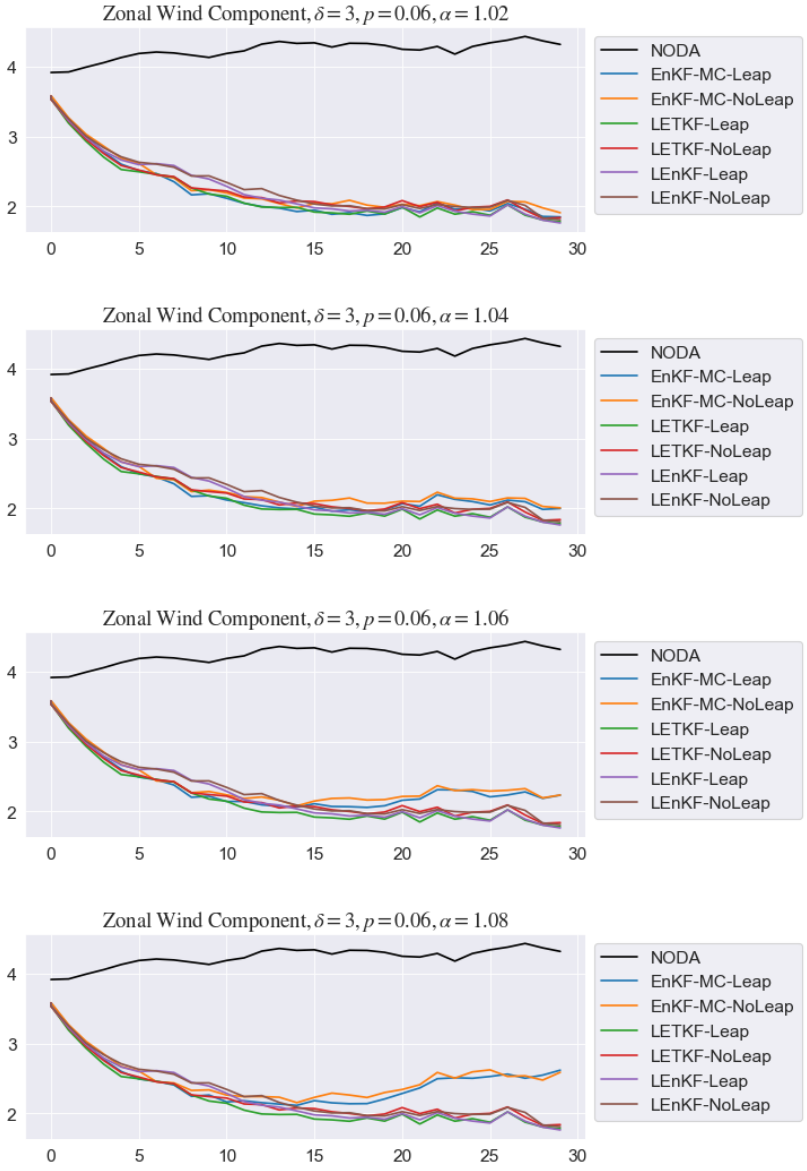


FIGURE 5.11: Time evolution of Zonal Wind Components for $\delta = 3$ and $p = 6\%$ varying α

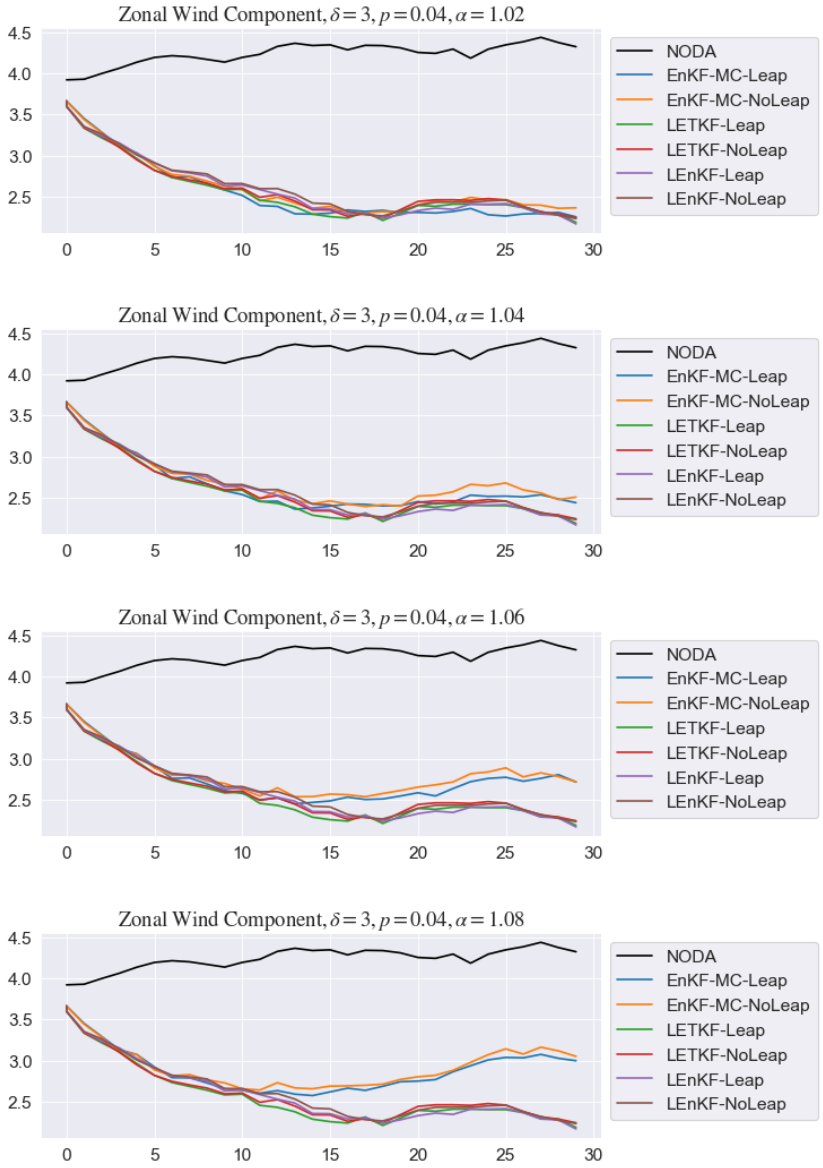


FIGURE 5.12: Time evolution of Zonal Wind Components for $\delta = 3$ and $p = 4\%$ varying α

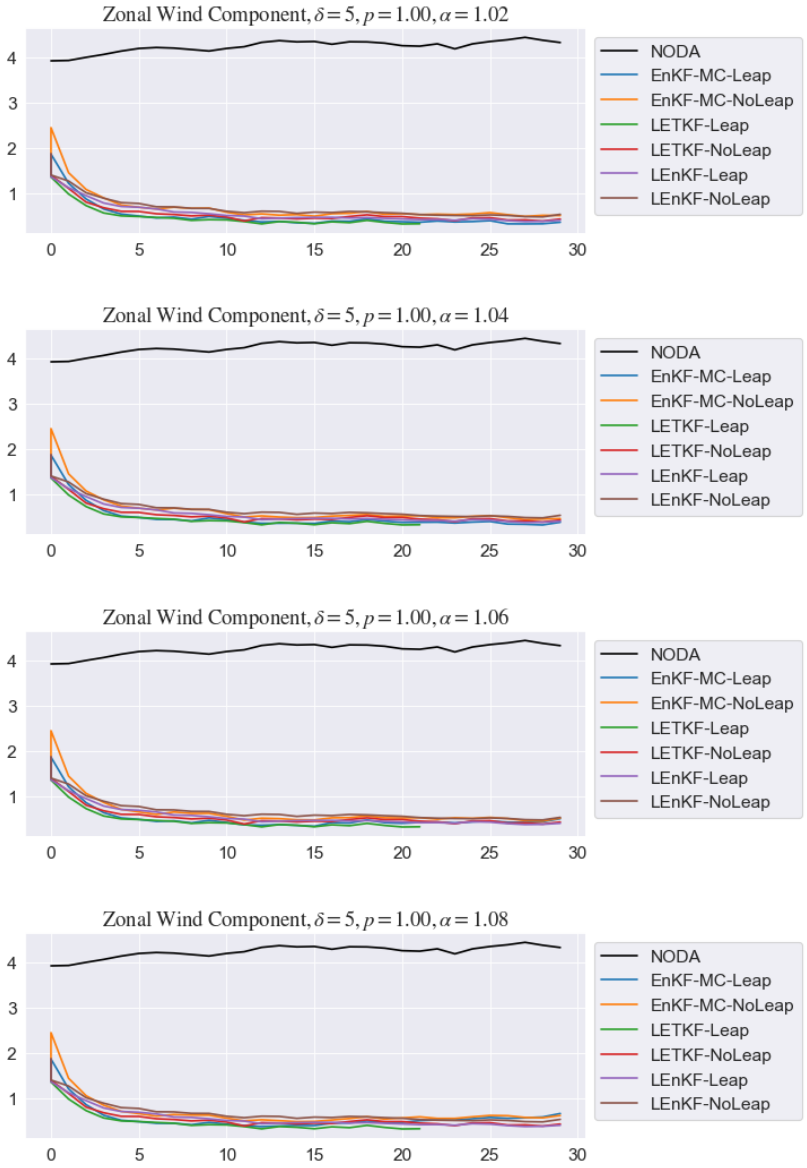


FIGURE 5.13: Time evolution of Zonal Wind Components for $\delta = 5$ and $p = 100\%$ varying α

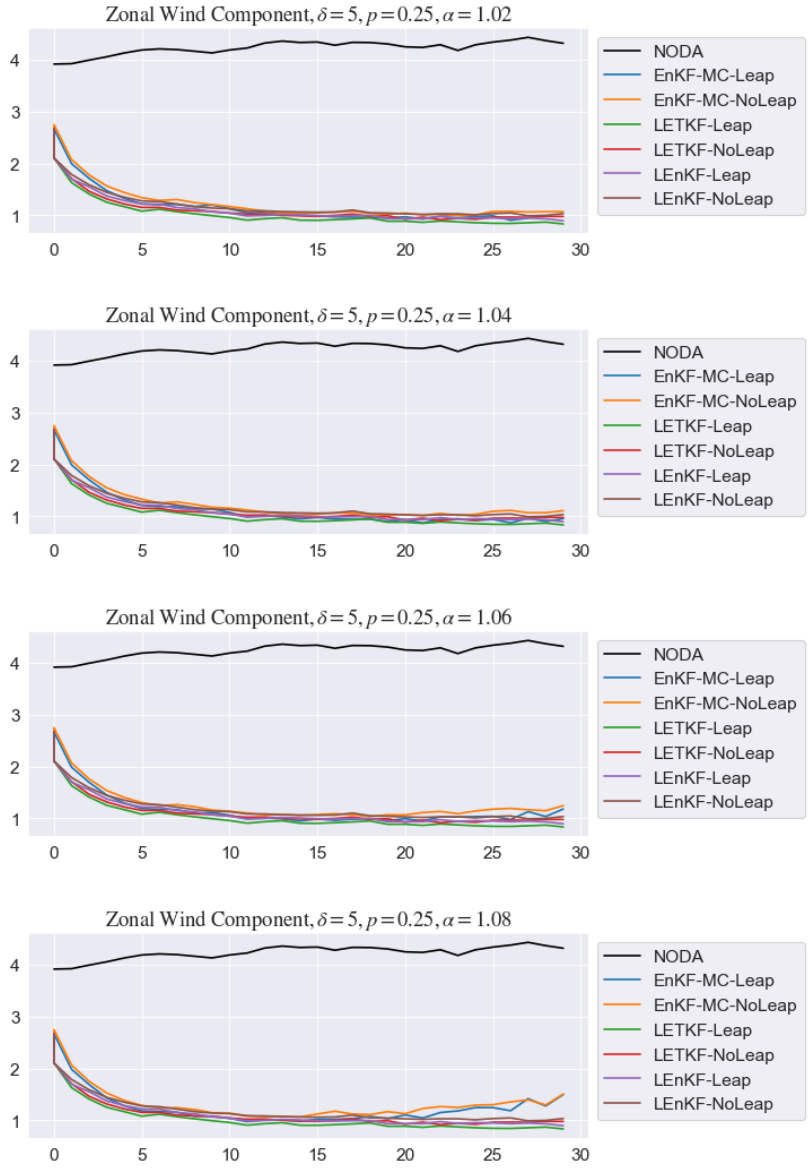


FIGURE 5.14: Time evolution of Zonal Wind Components for $\delta = 5$ and $p = 25\%$ varying α

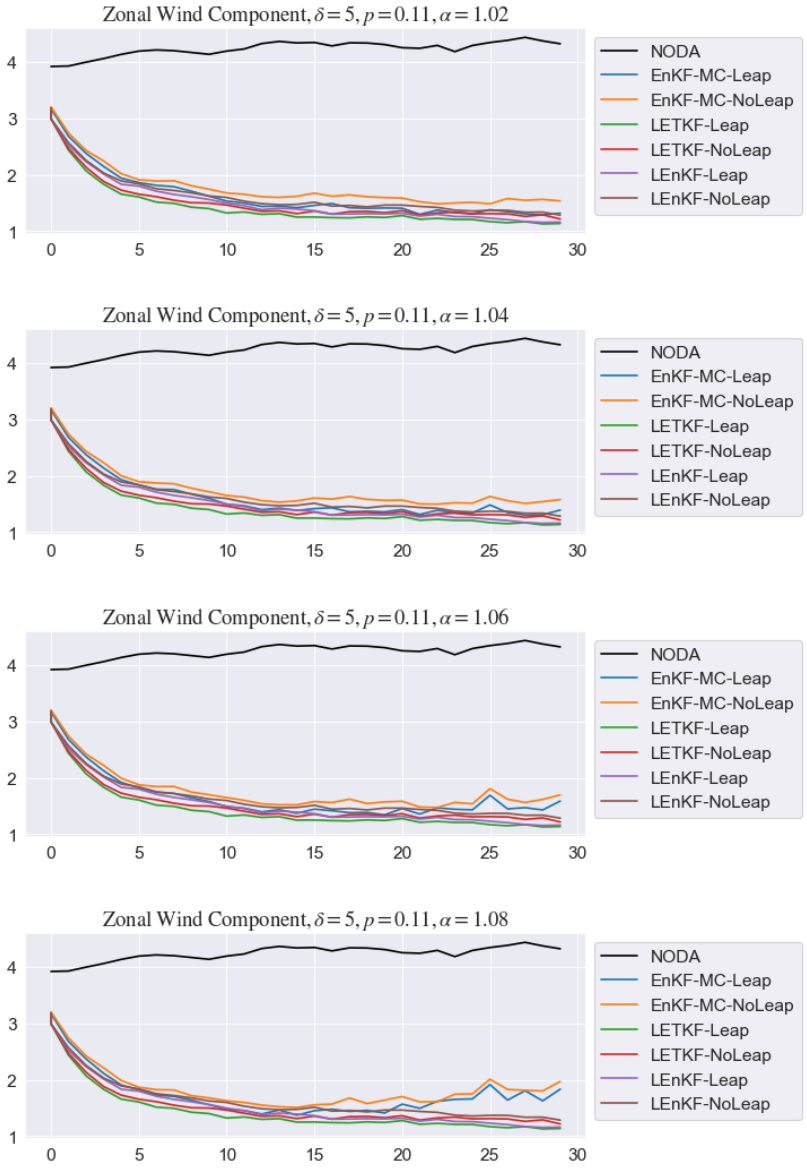


FIGURE 5.15: Time evolution of Zonal Wind Components for $\delta = 5$ and $p = 11\%$ varying α

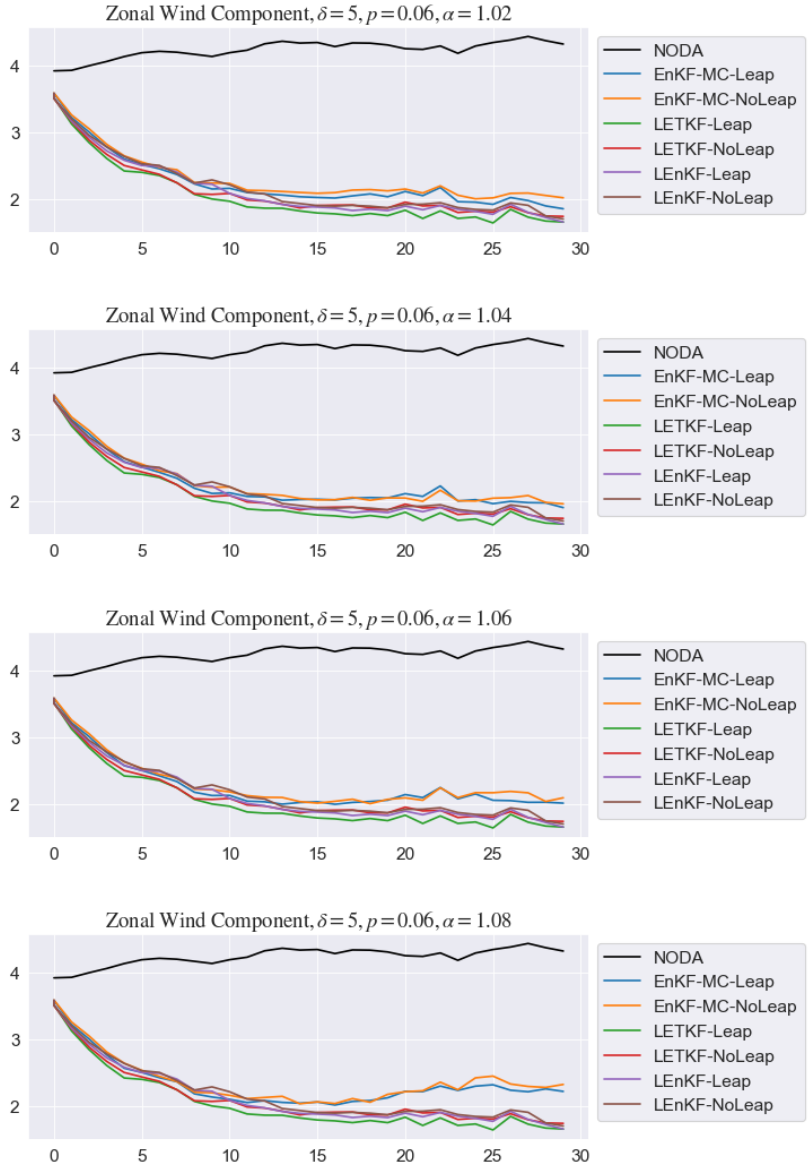


FIGURE 5.16: Time evolution of Zonal Wind Components for $\delta = 5$ and $p = 6\%$ varying α

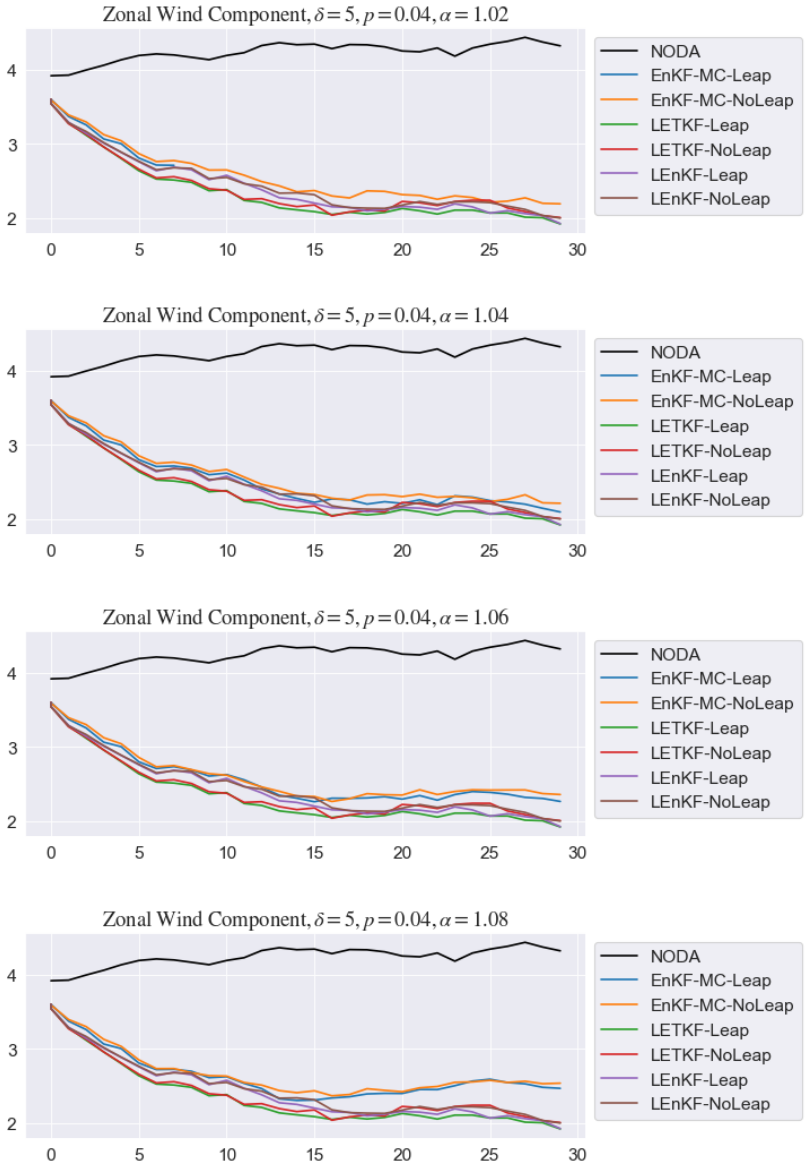


FIGURE 5.17: Time evolution of Zonal Wind Components for $\delta = 5$ and $p = 4\%$ varying α

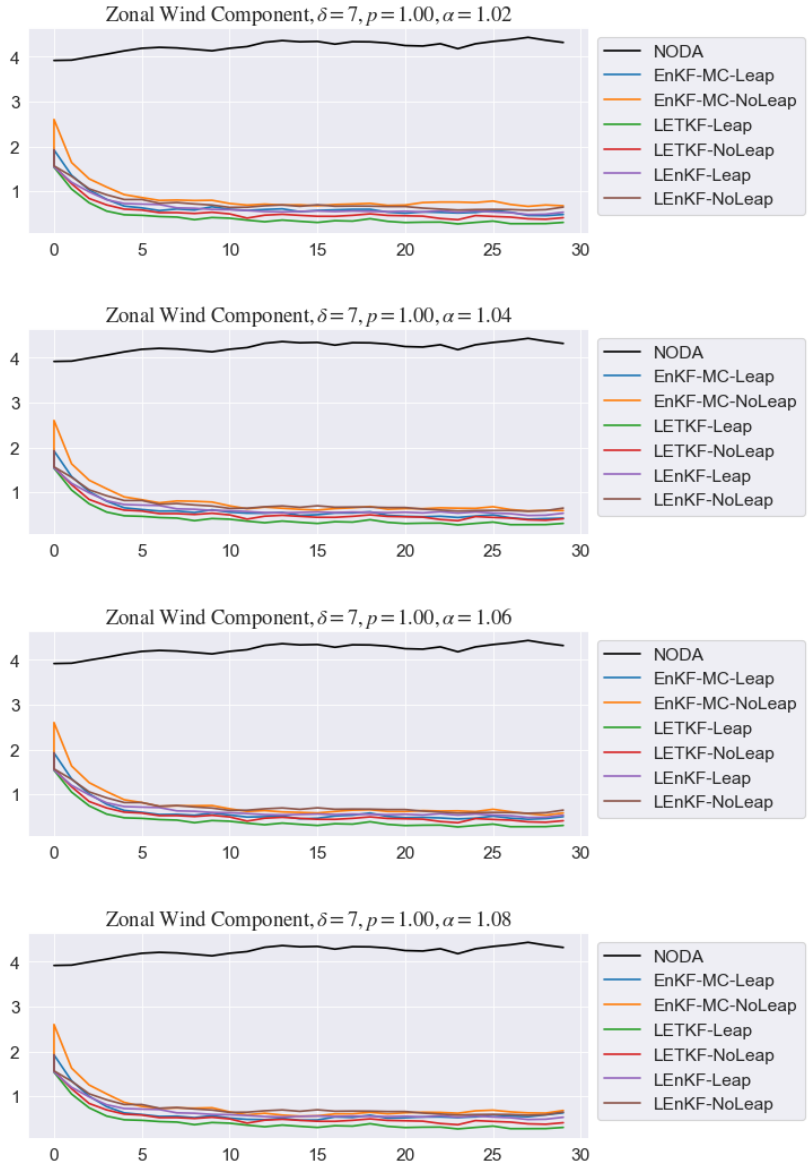


FIGURE 5.18: Time evolution of Zonal Wind Components for $\delta = 7$ and $p = 100\%$ varying α

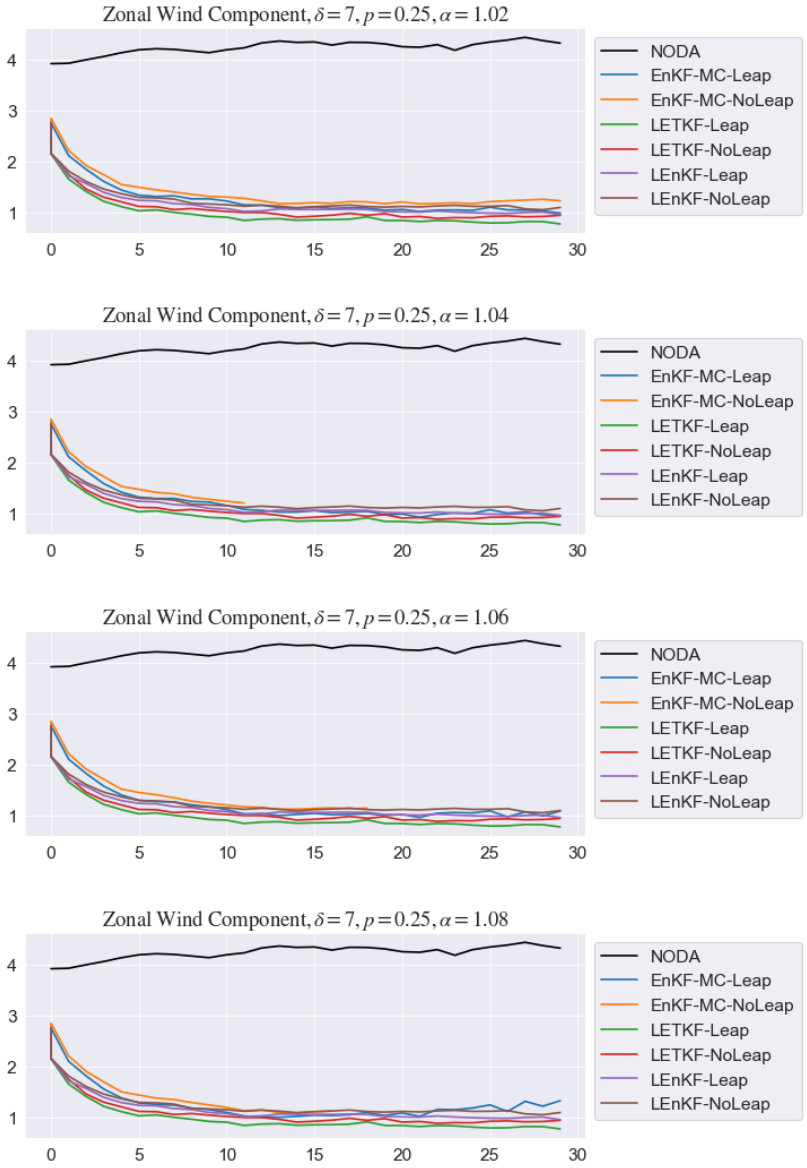


FIGURE 5.19: Time evolution of Zonal Wind Components for $\delta = 7$ and $p = 25\%$ varying α

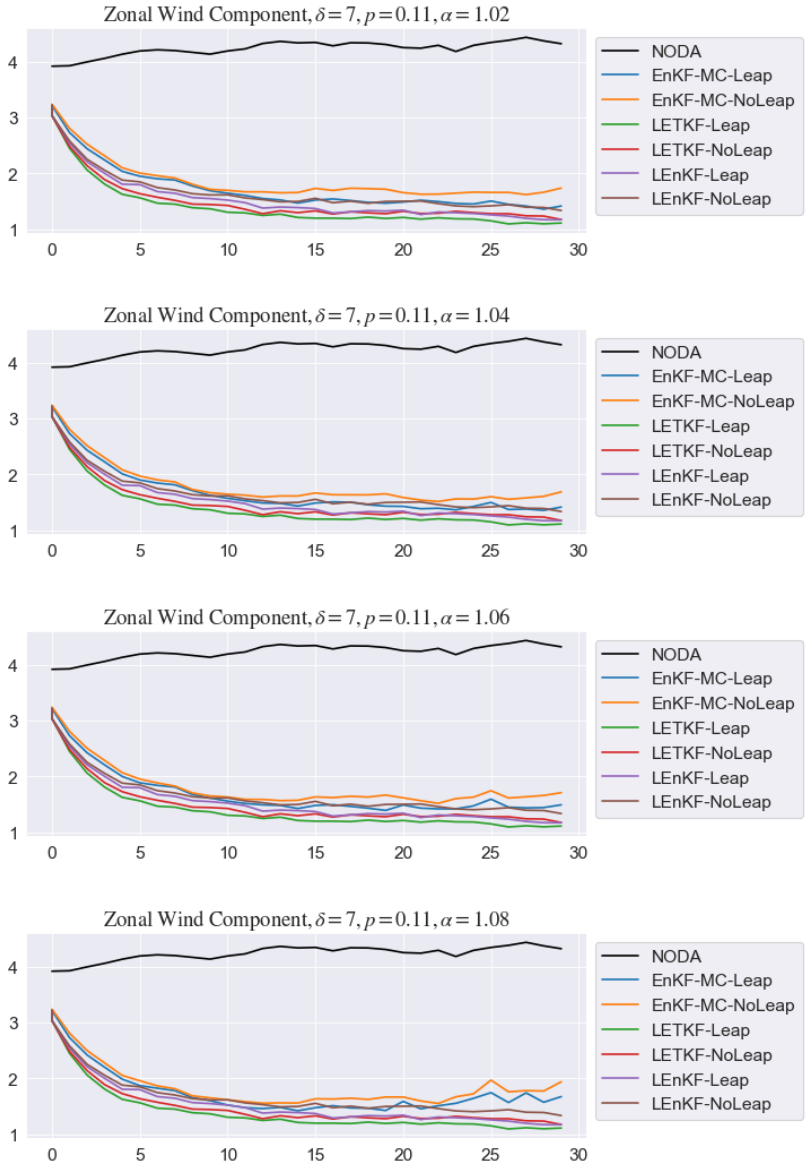


FIGURE 5.20: Time evolution of Zonal Wind Components for $\delta = 7$ and $p = 11\%$ varying α

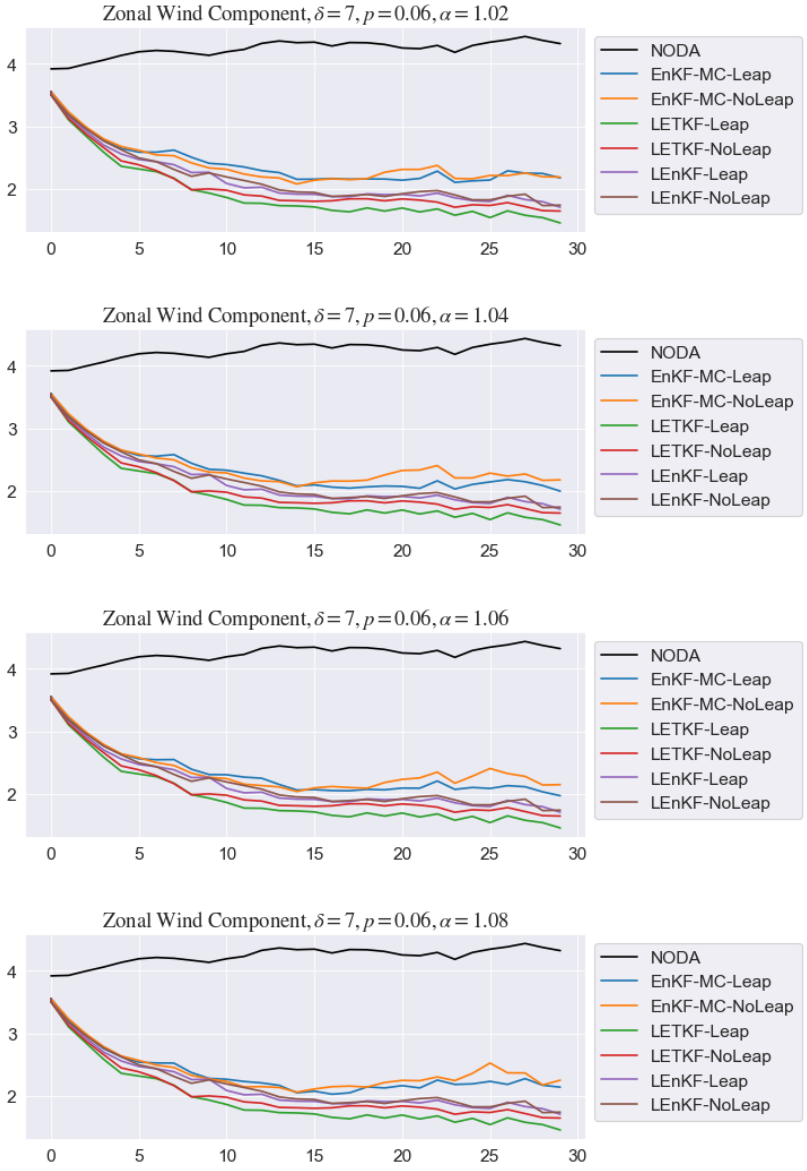


FIGURE 5.21: Time evolution of Zonal Wind Components for $\delta = 7$ and $p = 6\%$ varying α

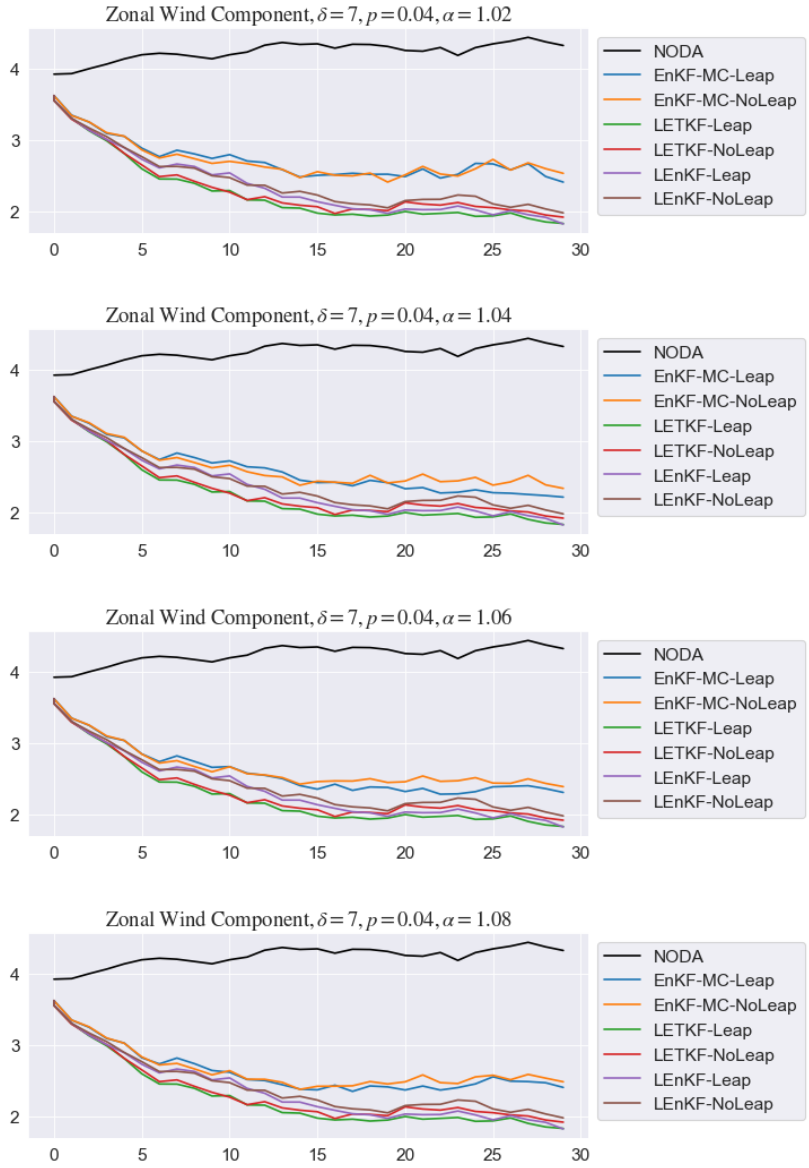


FIGURE 5.22: Time evolution of Zonal Wind Components for $\delta = 7$ and $p = 4\%$ varying α

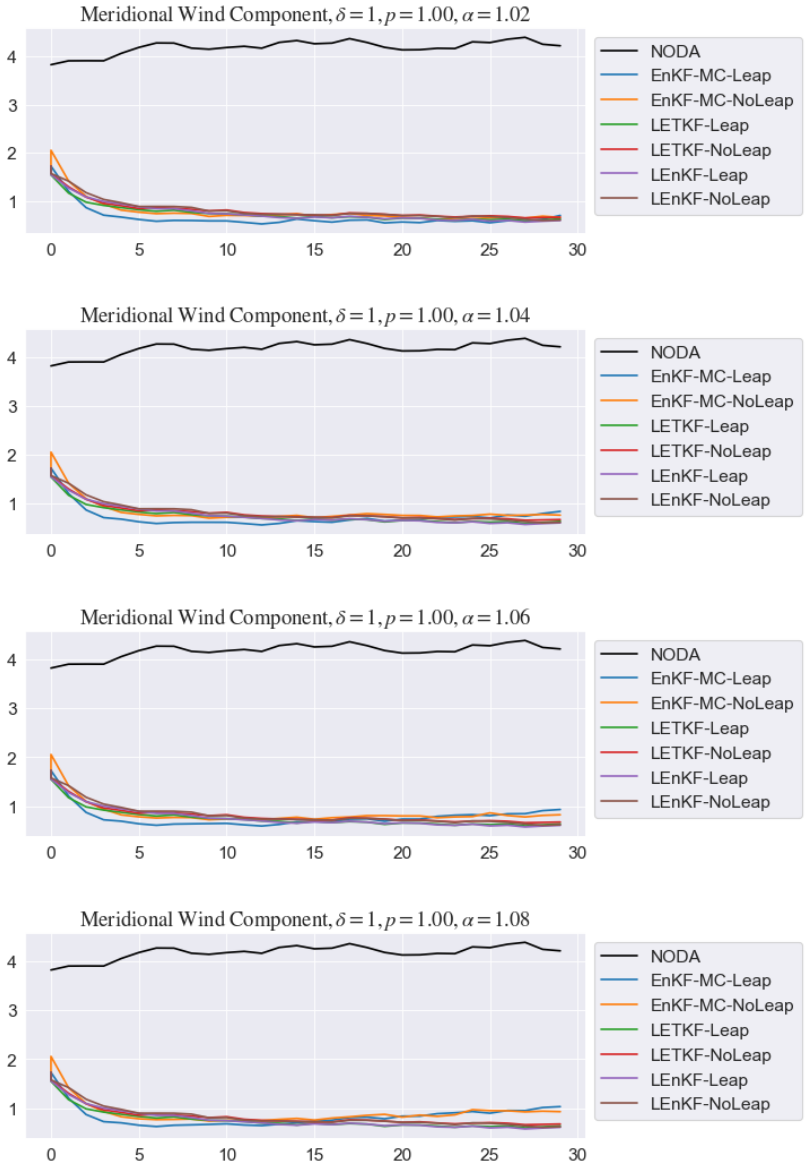


FIGURE 5.23: Time evolution of Meridional Wind Components for $\delta = 1$ and $p = 100\%$ varying α

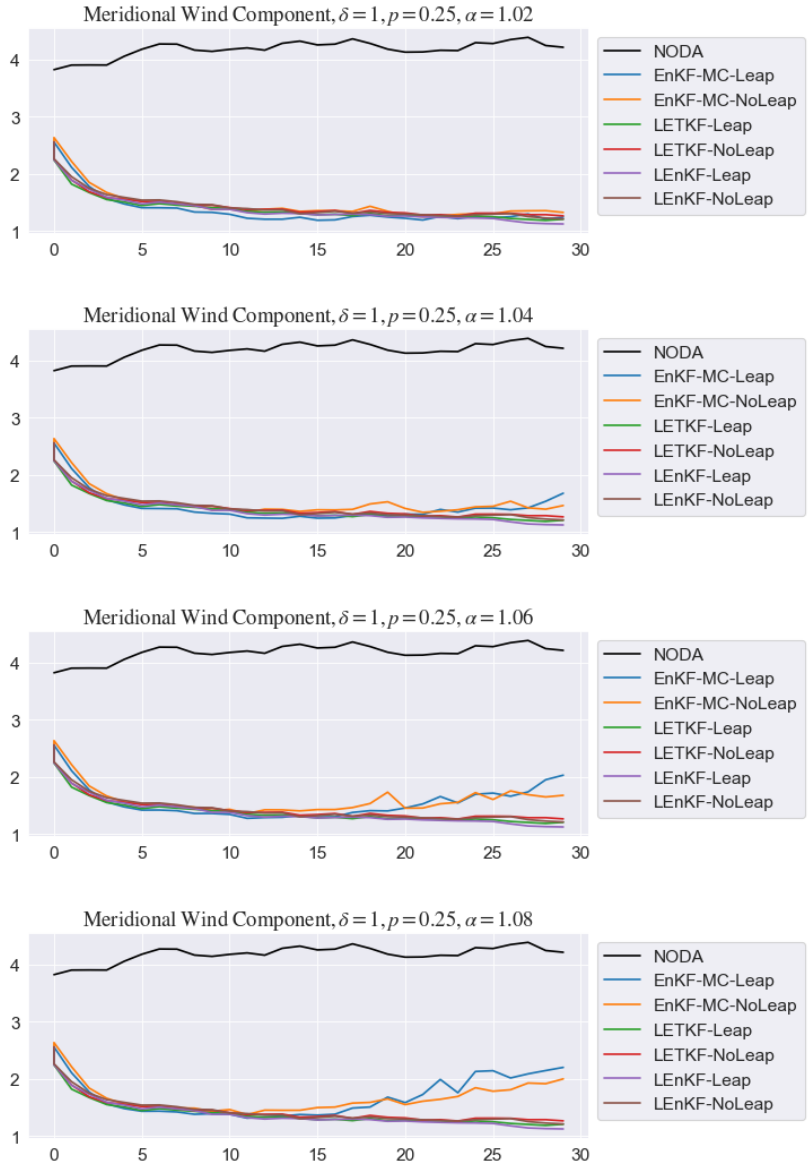


FIGURE 5.24: Time evolution of Meridional Wind Components for $\delta = 1$ and $p = 25\%$ varying α

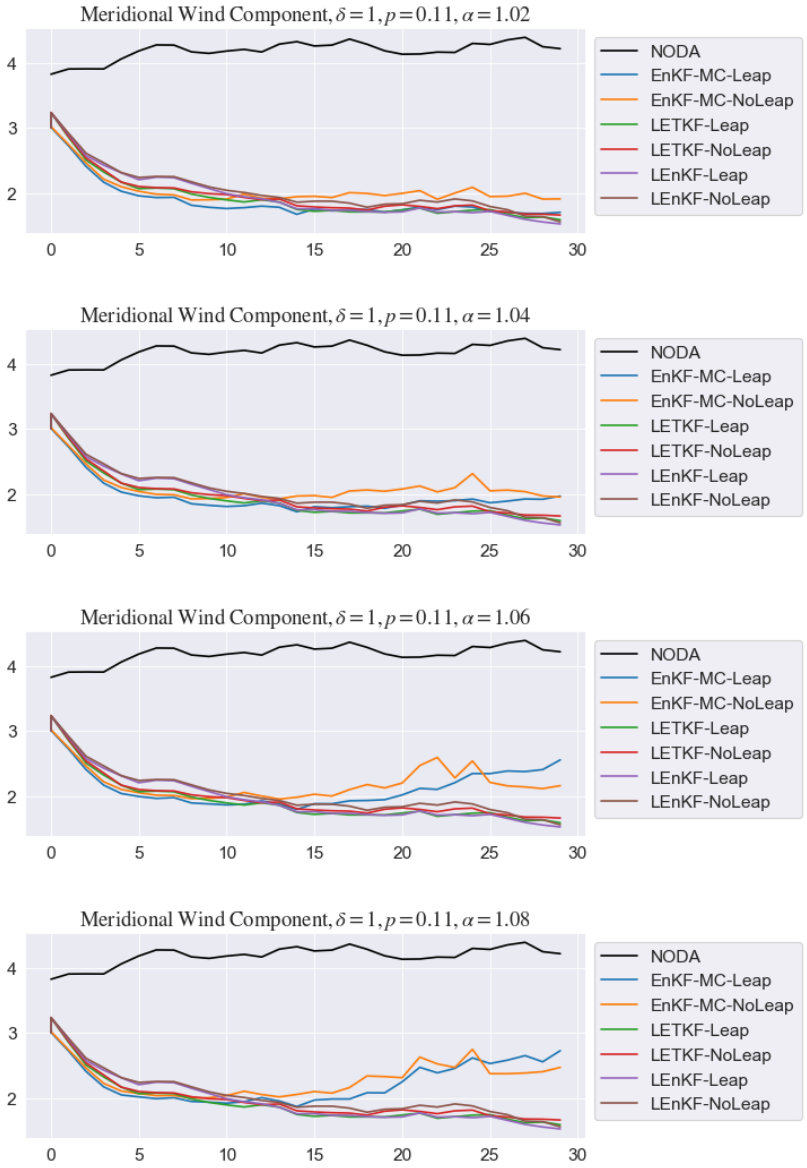


FIGURE 5.25: Time evolution of Meridional Wind Components for $\delta = 1$ and $p = 11\%$ varying α

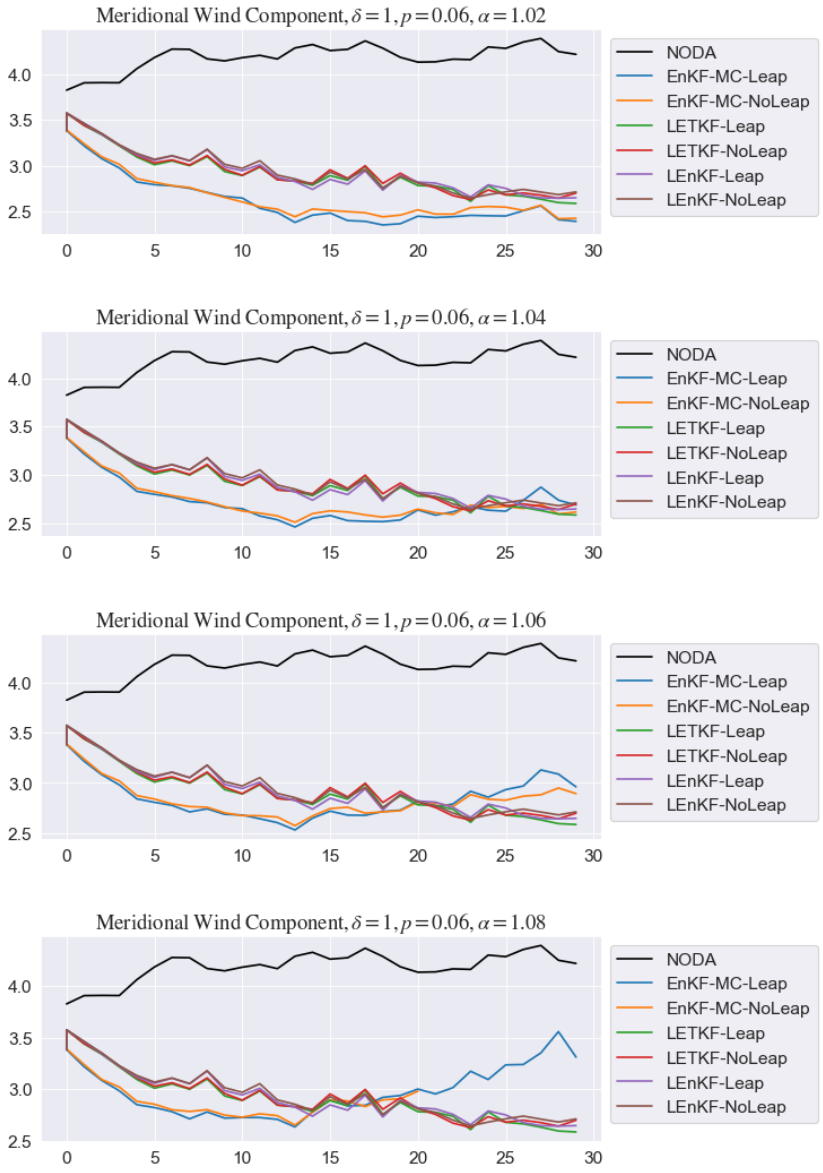


FIGURE 5.26: Time evolution of Meridional Wind Components for $\delta = 1$ and $p = 6\%$ varying α

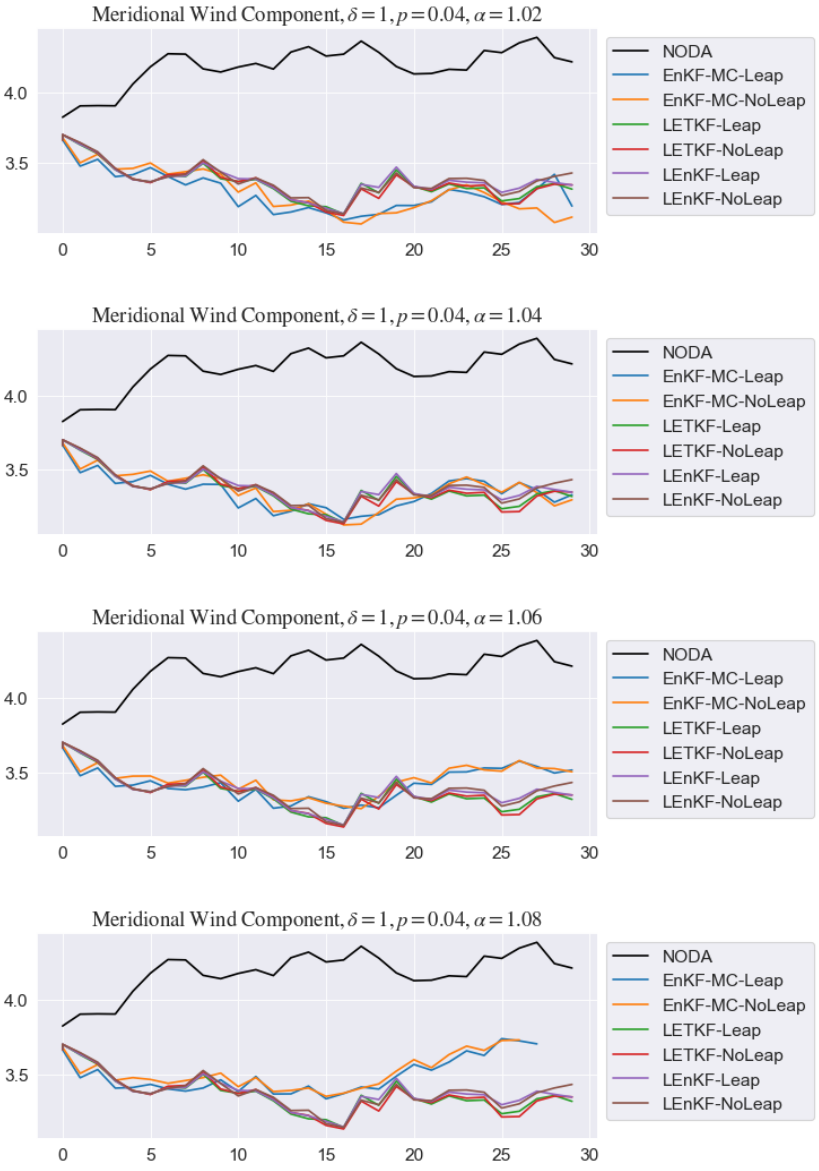


FIGURE 5.27: Time evolution of Meridional Wind Components for $\delta = 1$ and $p = 4\%$ varying α

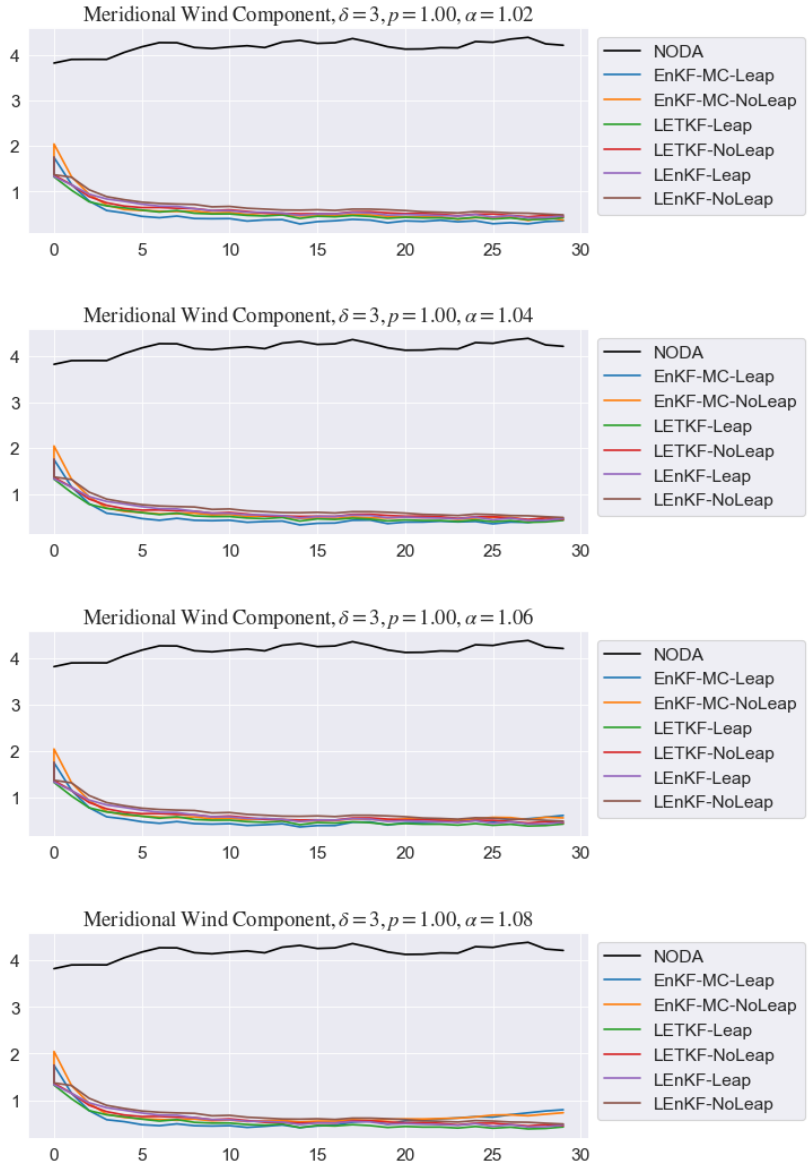


FIGURE 5.28: Time evolution of Meridional Wind Components for $\delta = 3$ and $p = 100\%$ varying α

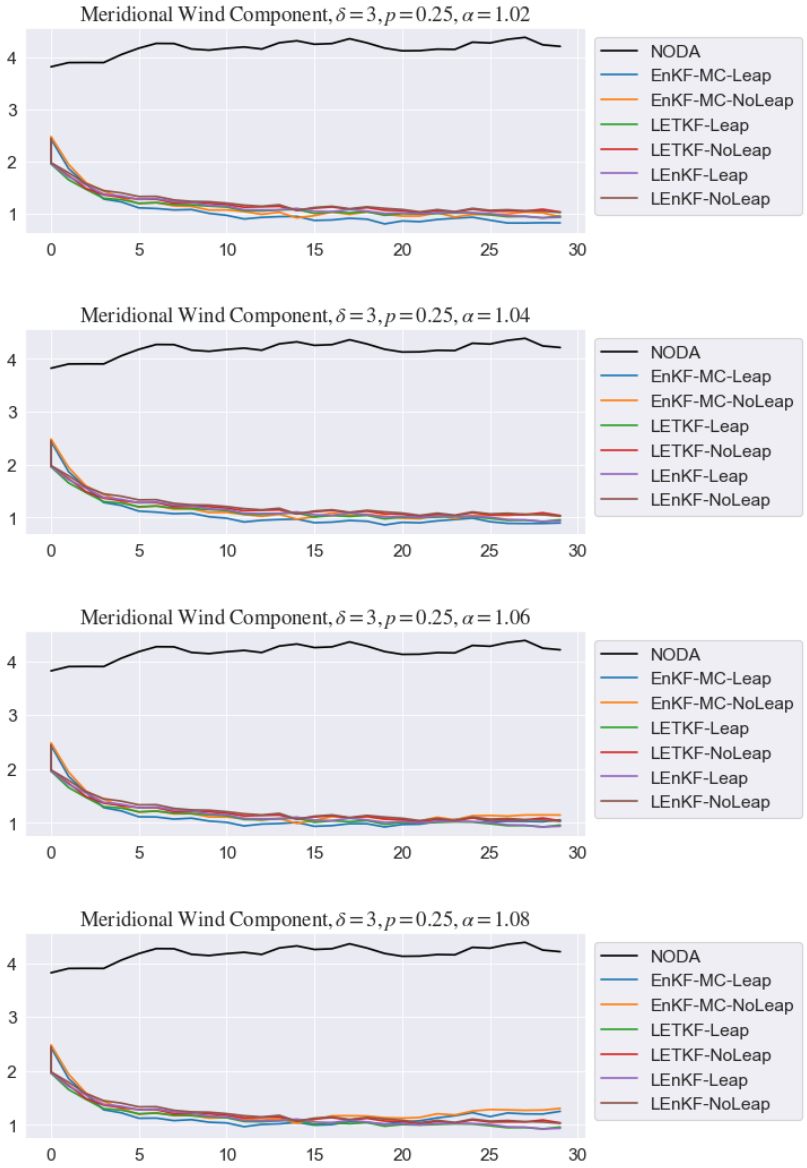


FIGURE 5.29: Time evolution of Meridional Wind Components for $\delta = 3$ and $p = 25\%$ varying α

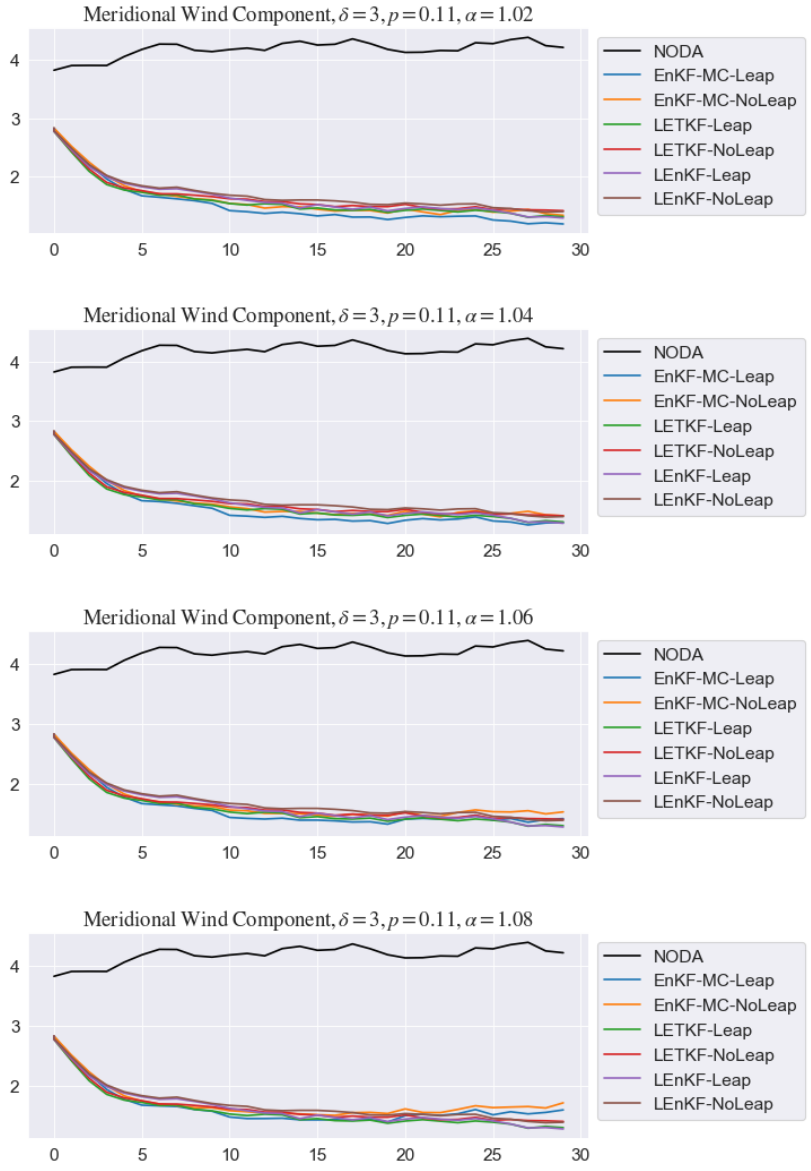


FIGURE 5.30: Time evolution of Meridional Wind Components for $\delta = 3$ and $p = 11\%$ varying α

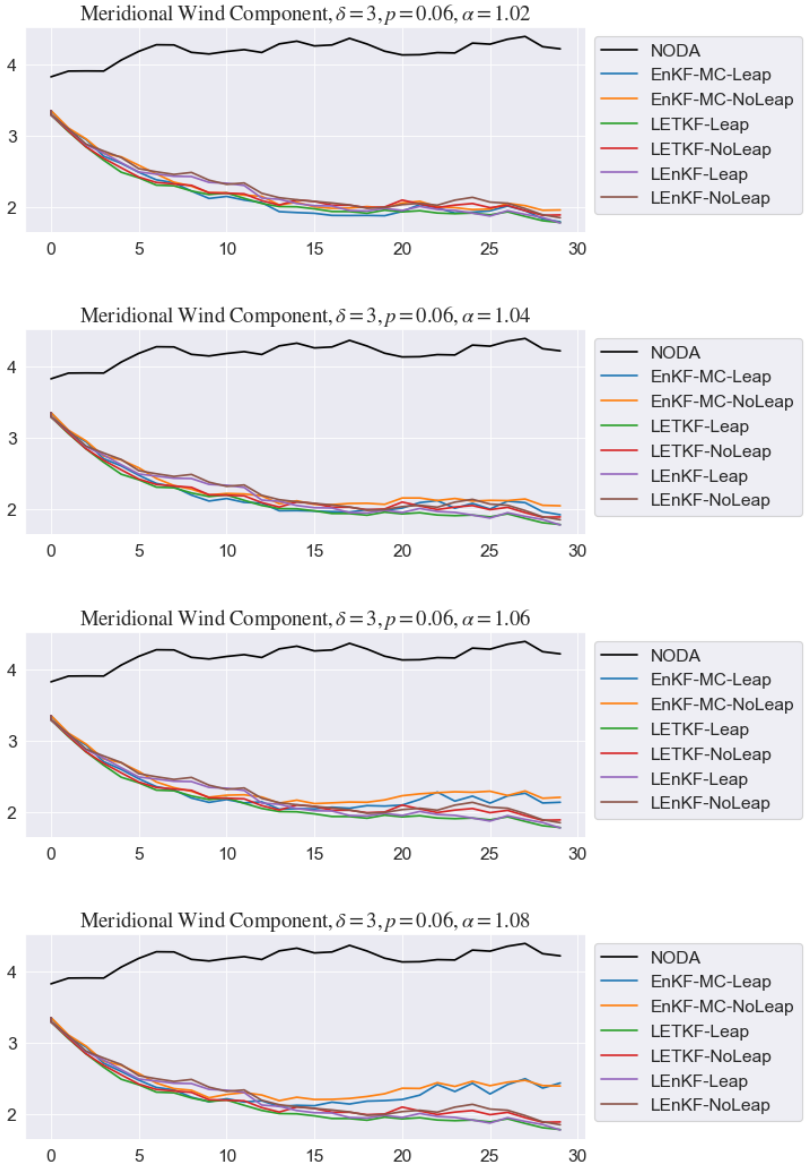


FIGURE 5.31: Time evolution of Meridional Wind Components for $\delta = 3$ and $p = 6\%$ varying α

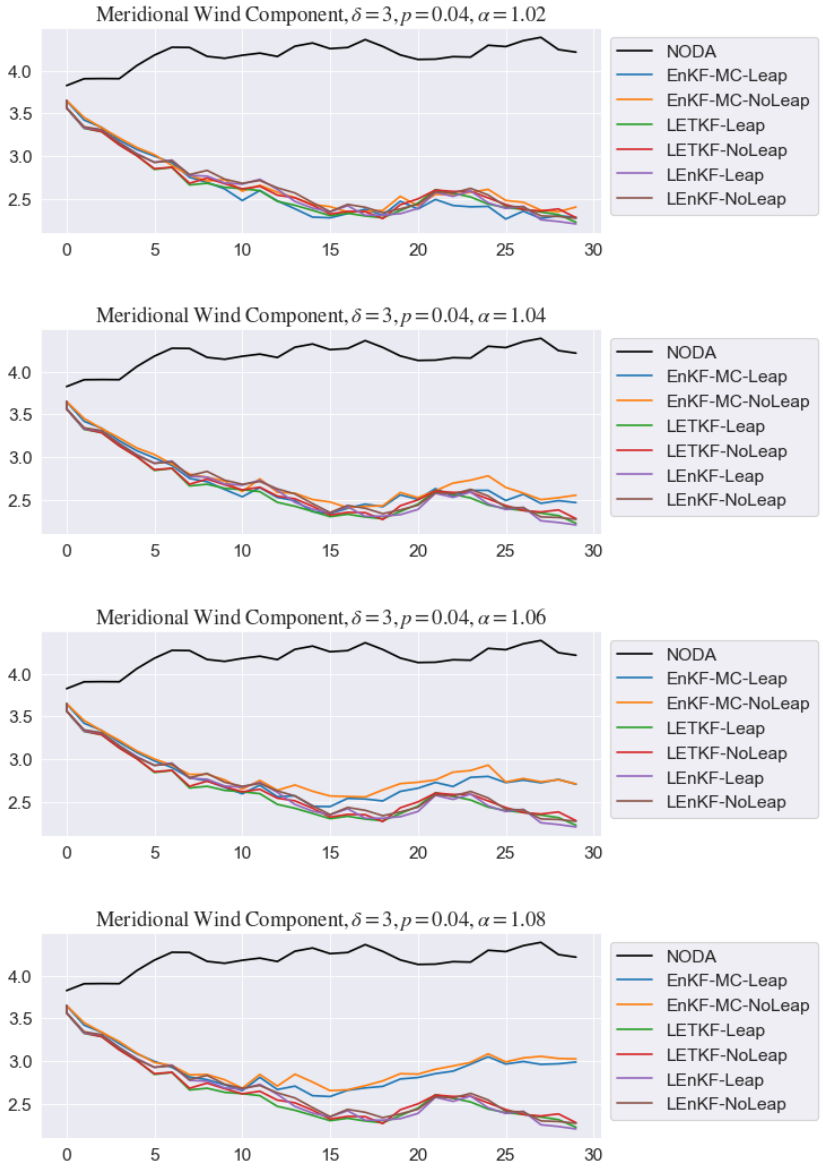


FIGURE 5.32: Time evolution of Meridional Wind Components for $\delta = 3$ and $p = 4\%$ varying α

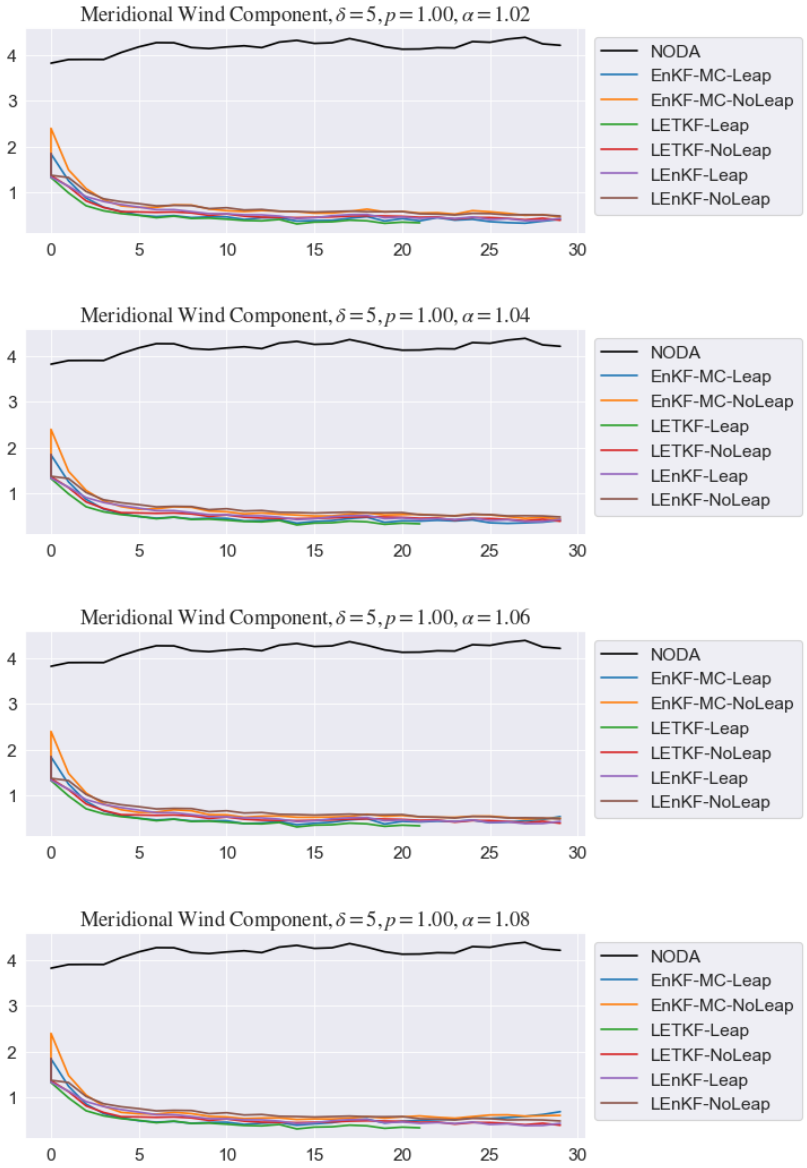


FIGURE 5.33: Time evolution of Meridional Wind Components for $\delta = 5$ and $p = 100\%$ varying α

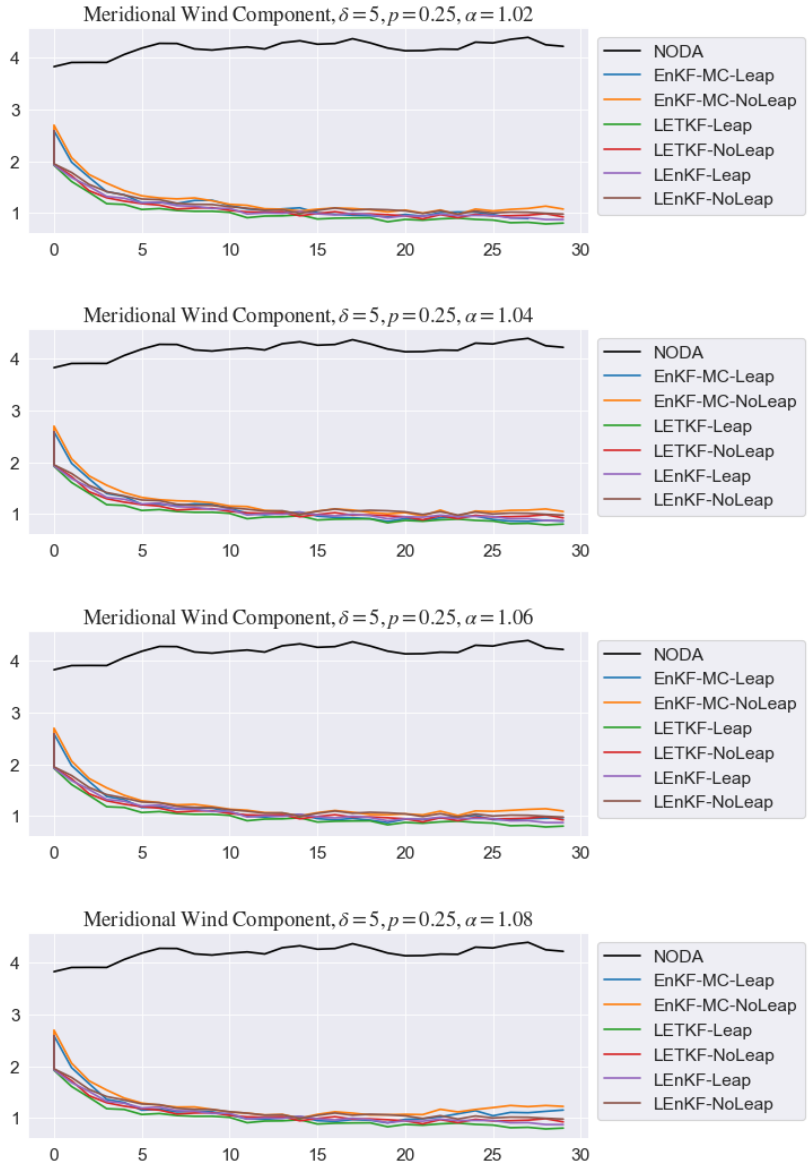


FIGURE 5.34: Time evolution of Meridional Wind Components for $\delta = 5$ and $p = 25\%$ varying α

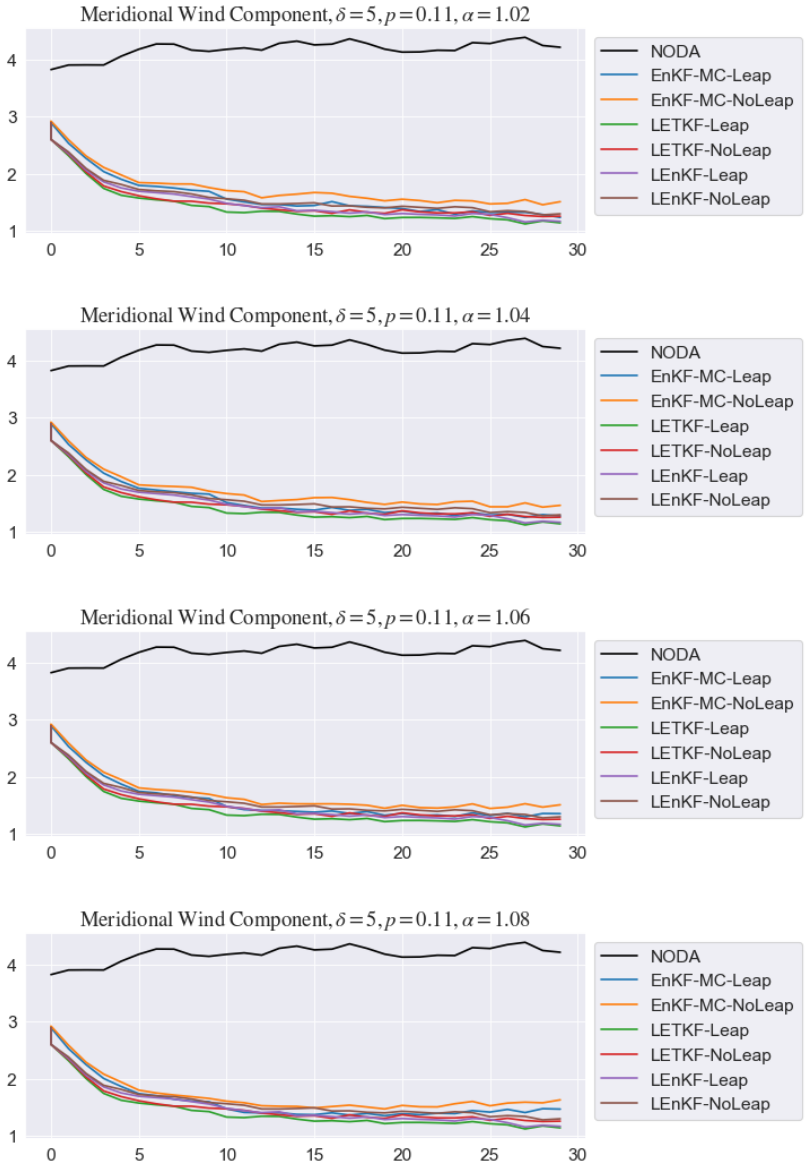


FIGURE 5.35: Time evolution of Meridional Wind Components for $\delta = 5$ and $p = 11\%$ varying α

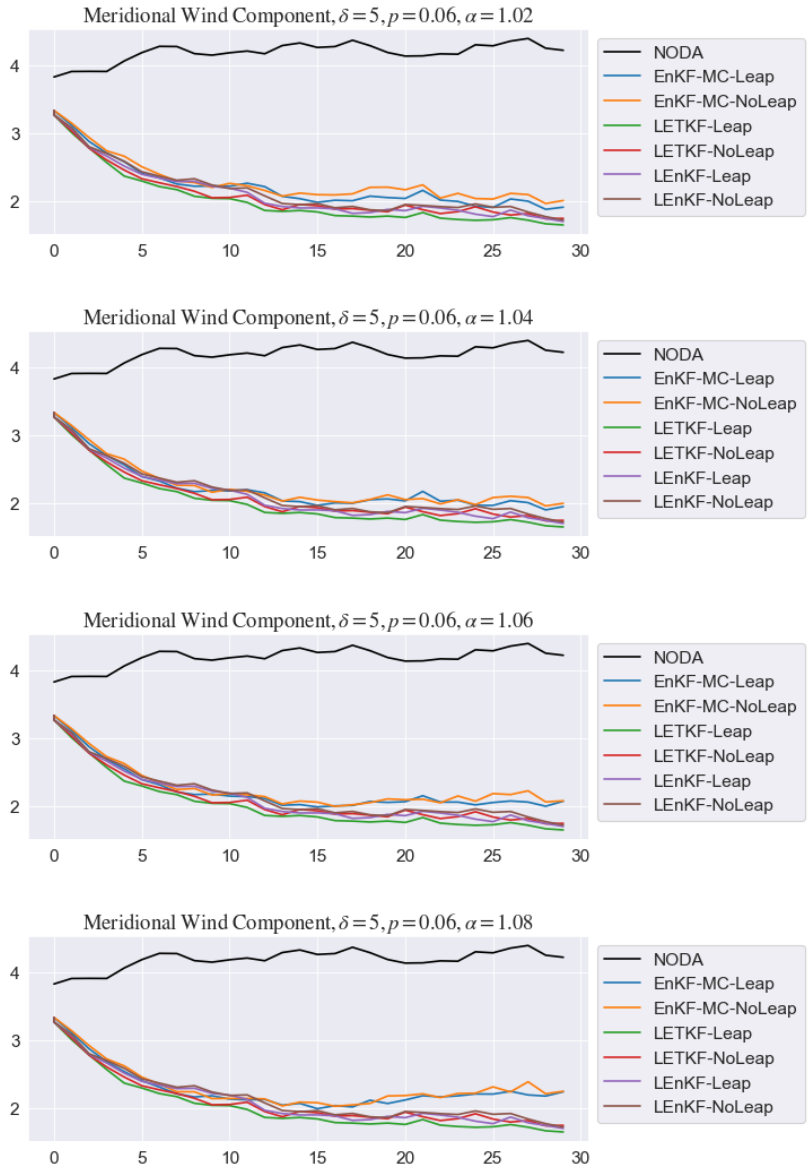


FIGURE 5.36: Time evolution of Meridional Wind Components for $\delta = 5$ and $p = 6\%$ varying α

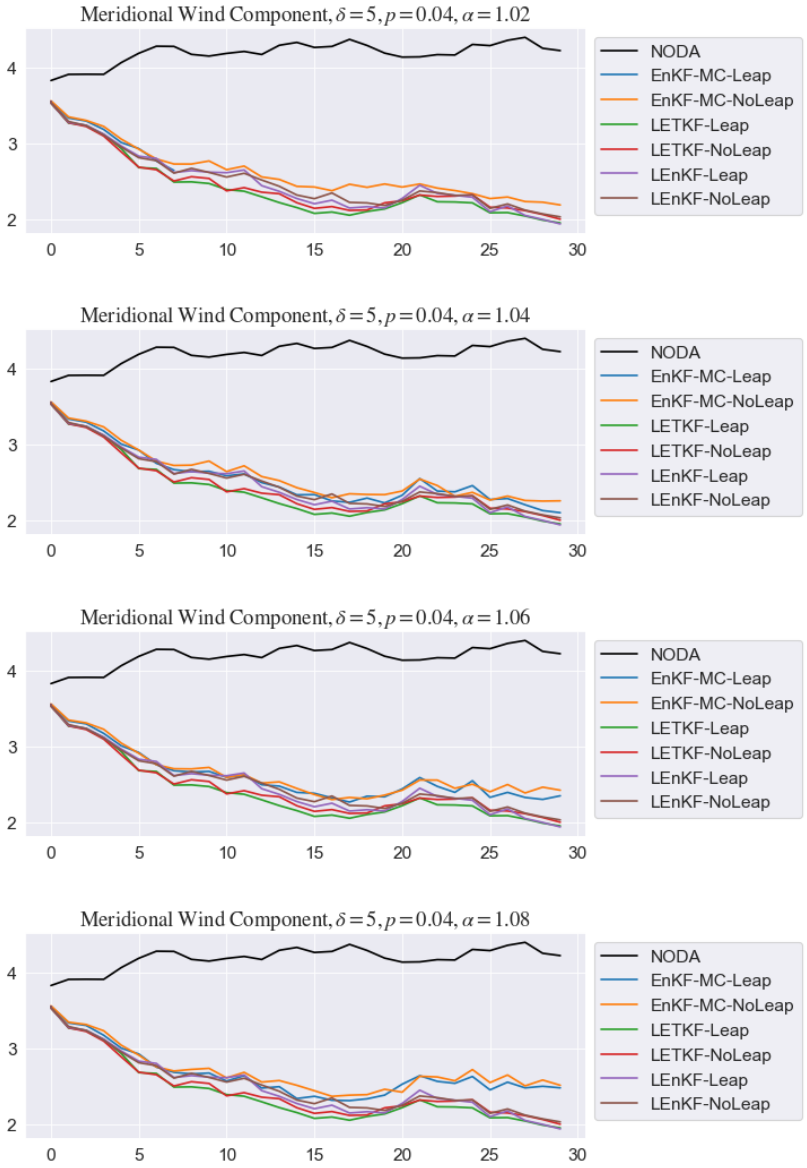


FIGURE 5.37: Time evolution of Meridional Wind Components for $\delta = 5$ and $p = 4\%$ varying α

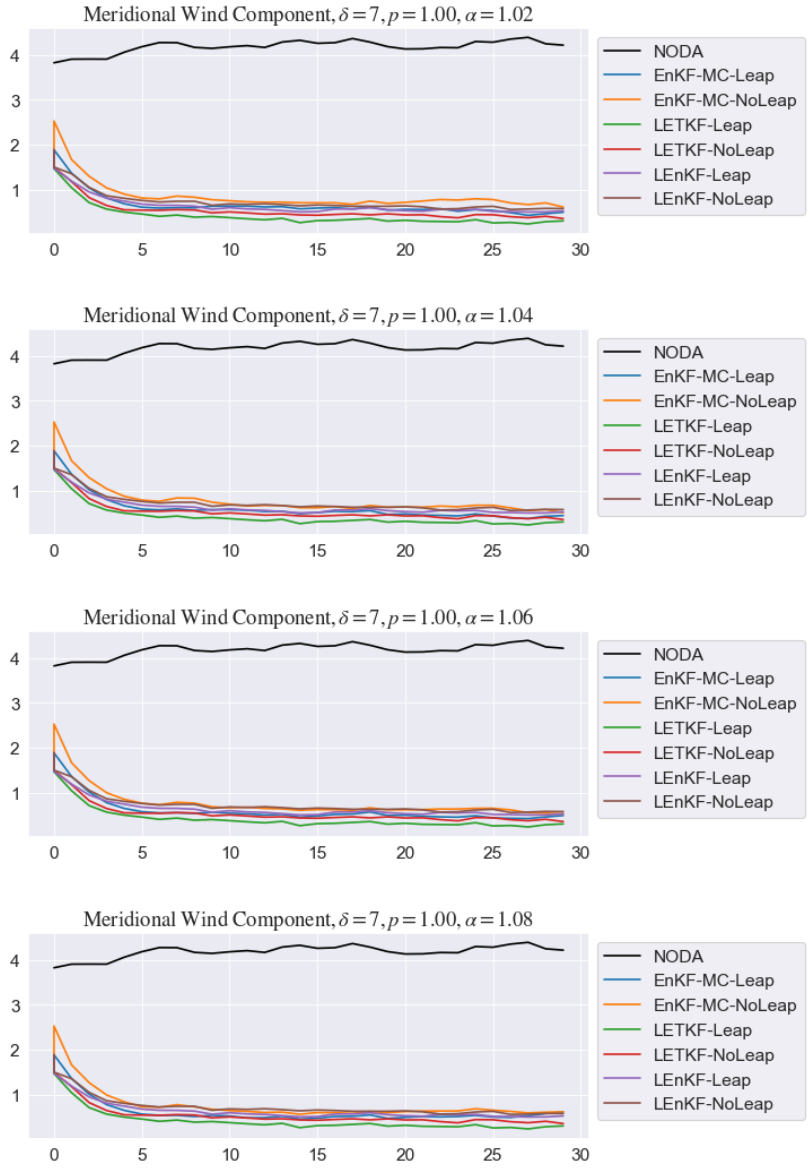


FIGURE 5.38: Time evolution of Meridional Wind Components for $\delta = 7$ and $p = 100\%$ varying α

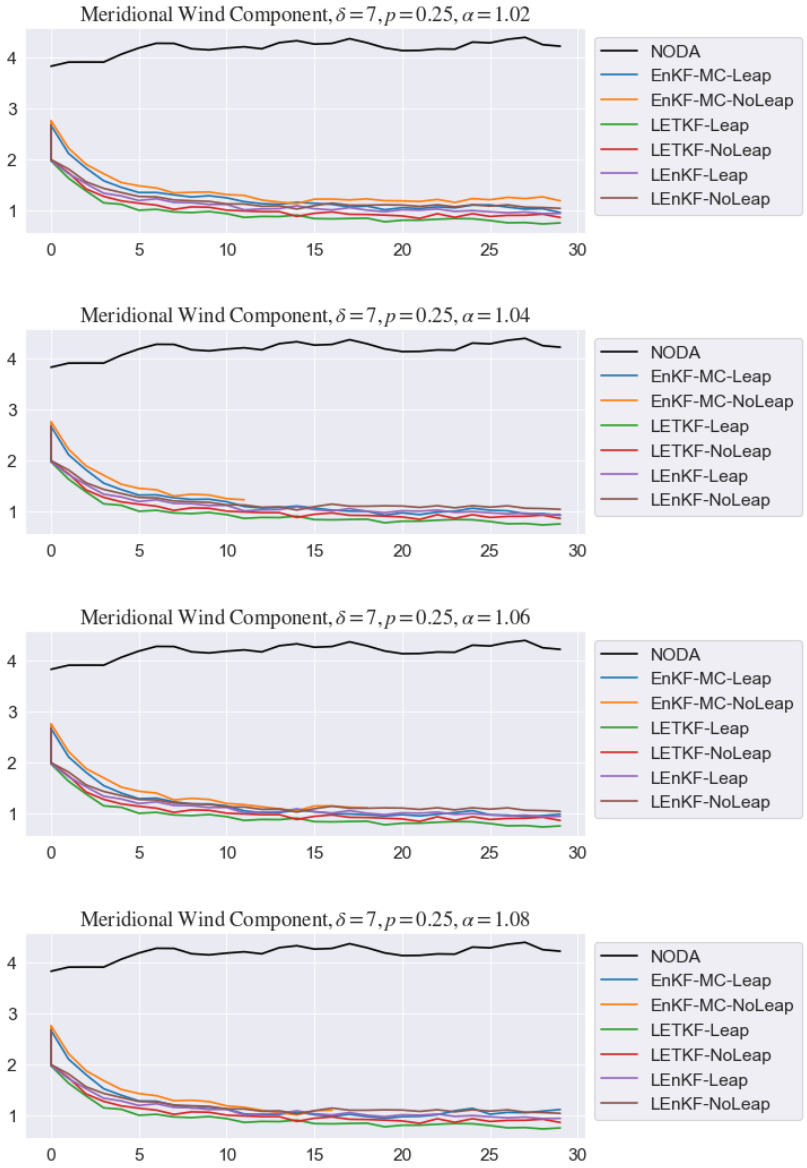


FIGURE 5.39: Time evolution of Meridional Wind Components for $\delta = 7$ and $p = 25\%$ varying α

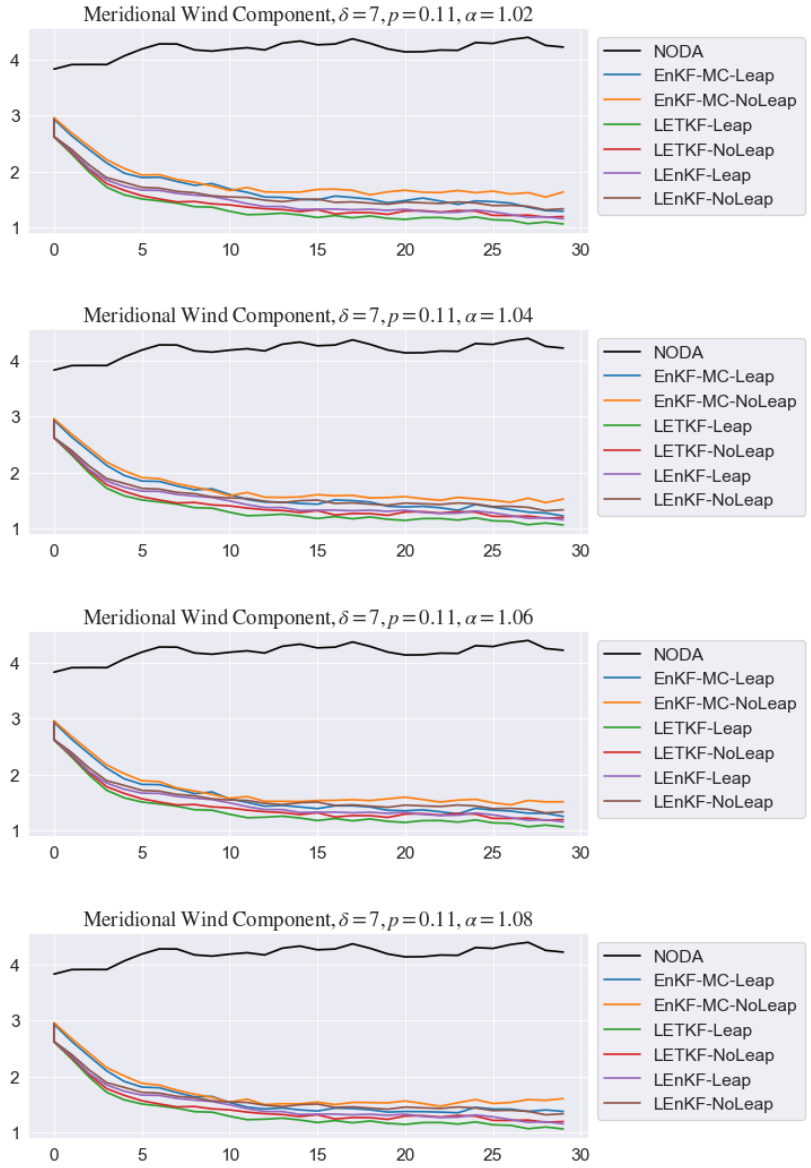


FIGURE 5.40: Time evolution of Meridional Wind Components for $\delta = 7$ and $p = 11\%$ varying α

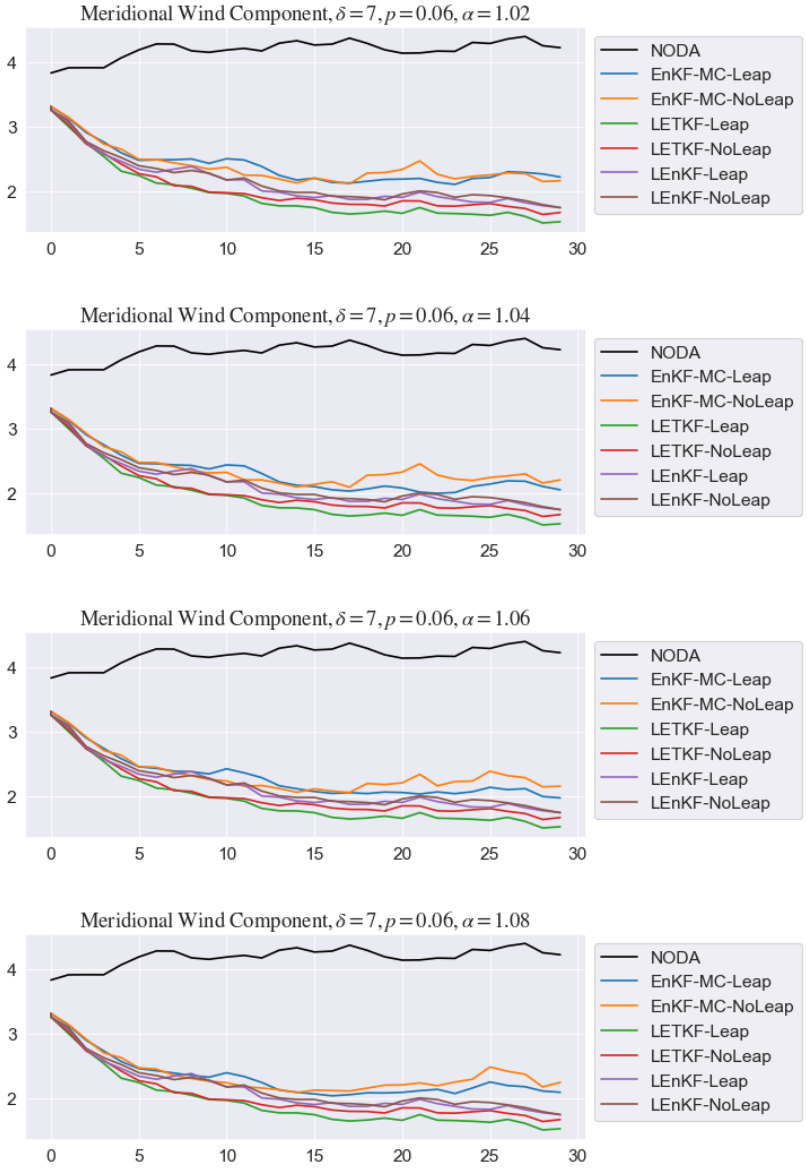


FIGURE 5.41: Time evolution of Meridional Wind Components for $\delta = 7$ and $p = 6\%$ varying α

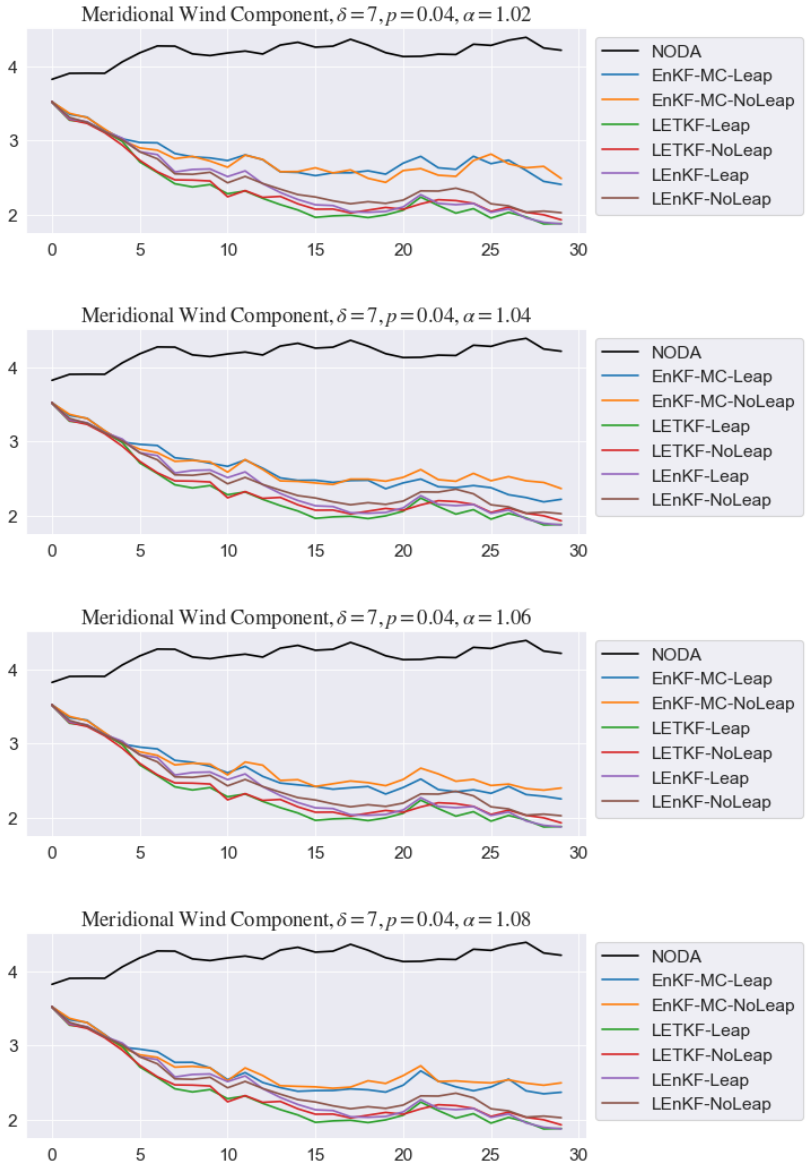


FIGURE 5.42: Time evolution of Meridional Wind Components for $\delta = 7$ and $p = 4\%$ varying α

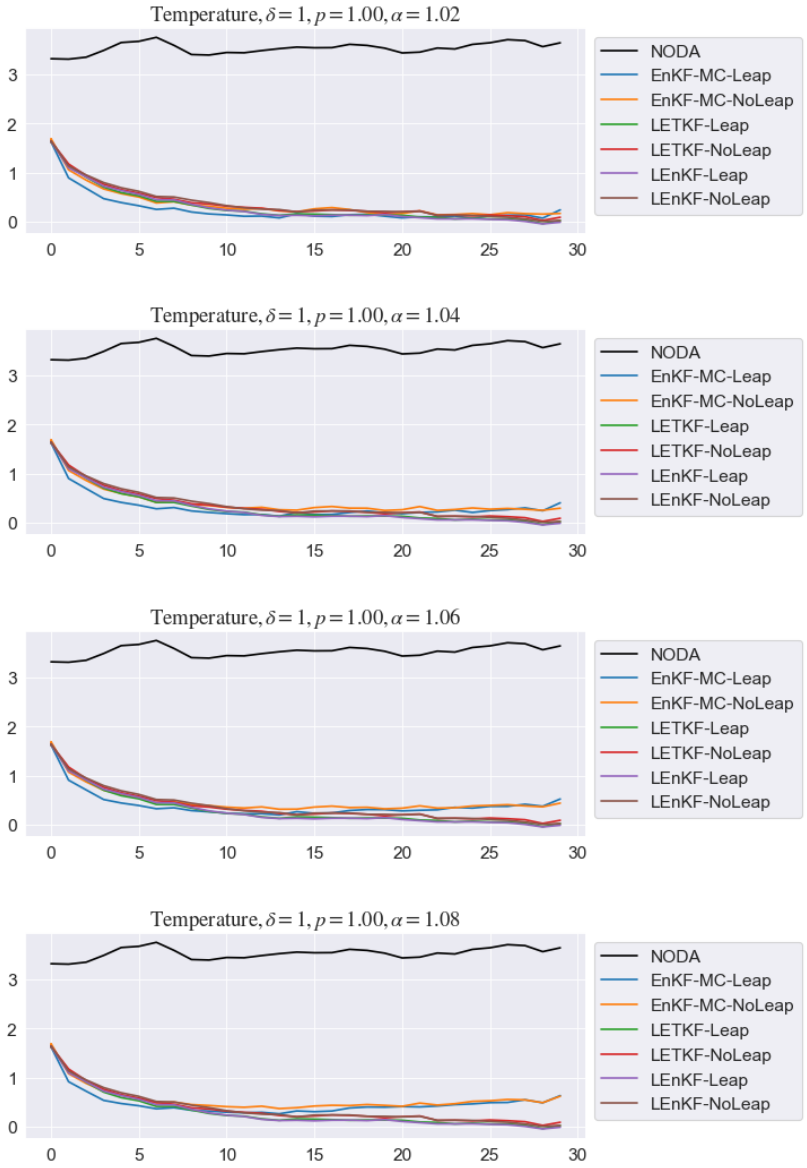


FIGURE 5.43: Time evolution of Temperature for $\delta = 1$ and $p = 100\%$ varying α

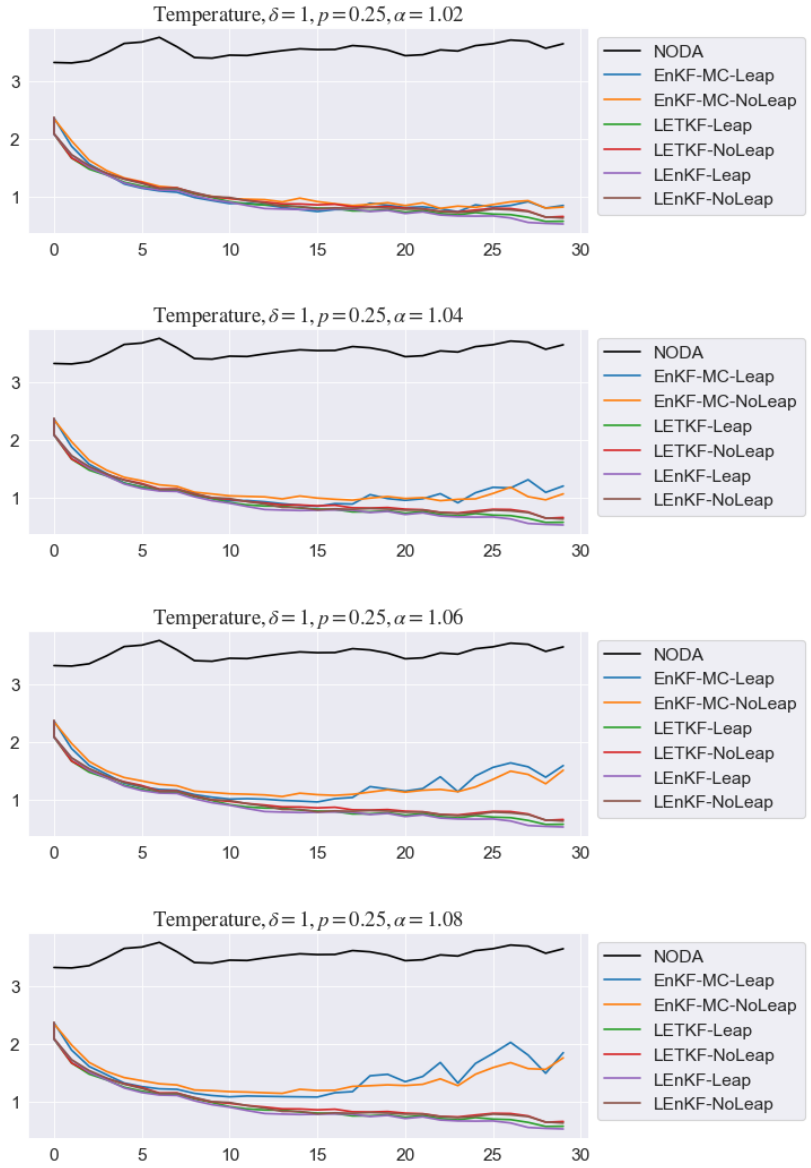


FIGURE 5.44: Time evolution of Temperature for $\delta = 1$ and $p = 25\%$ varying α

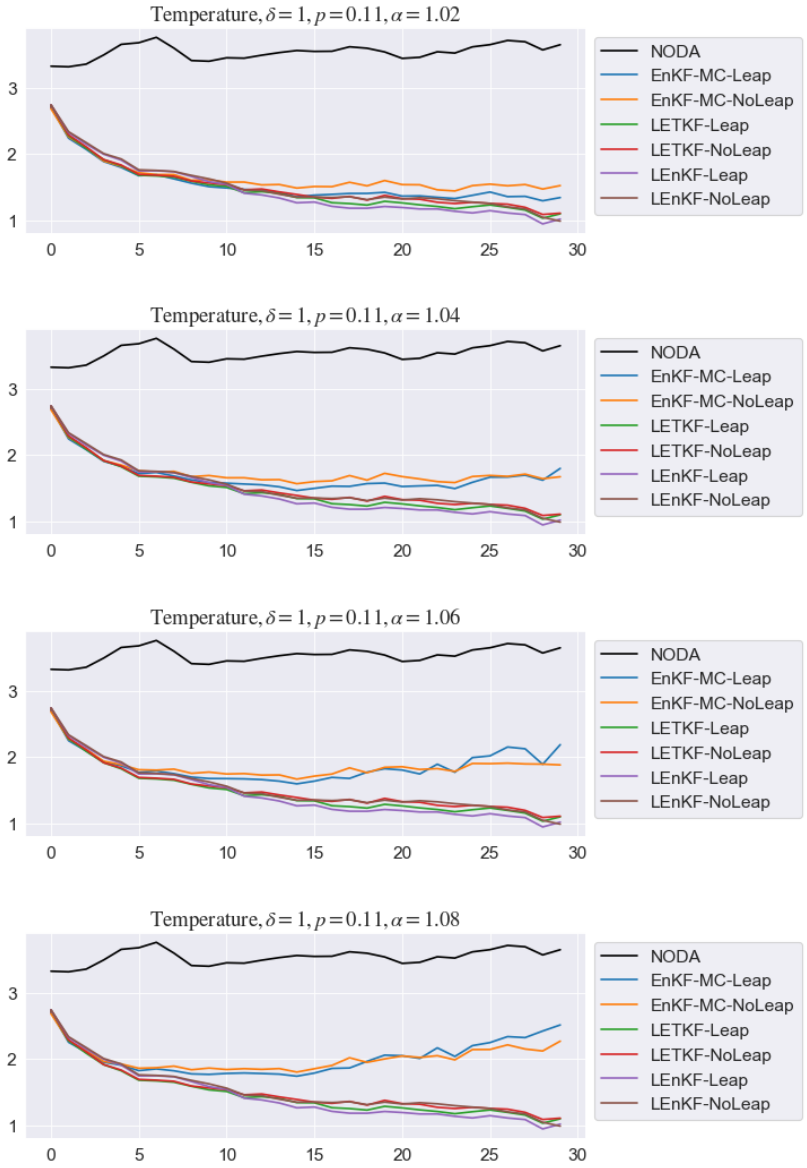


FIGURE 5.45: Time evolution of Temperature for $\delta = 1$ and $p = 11\%$ varying α

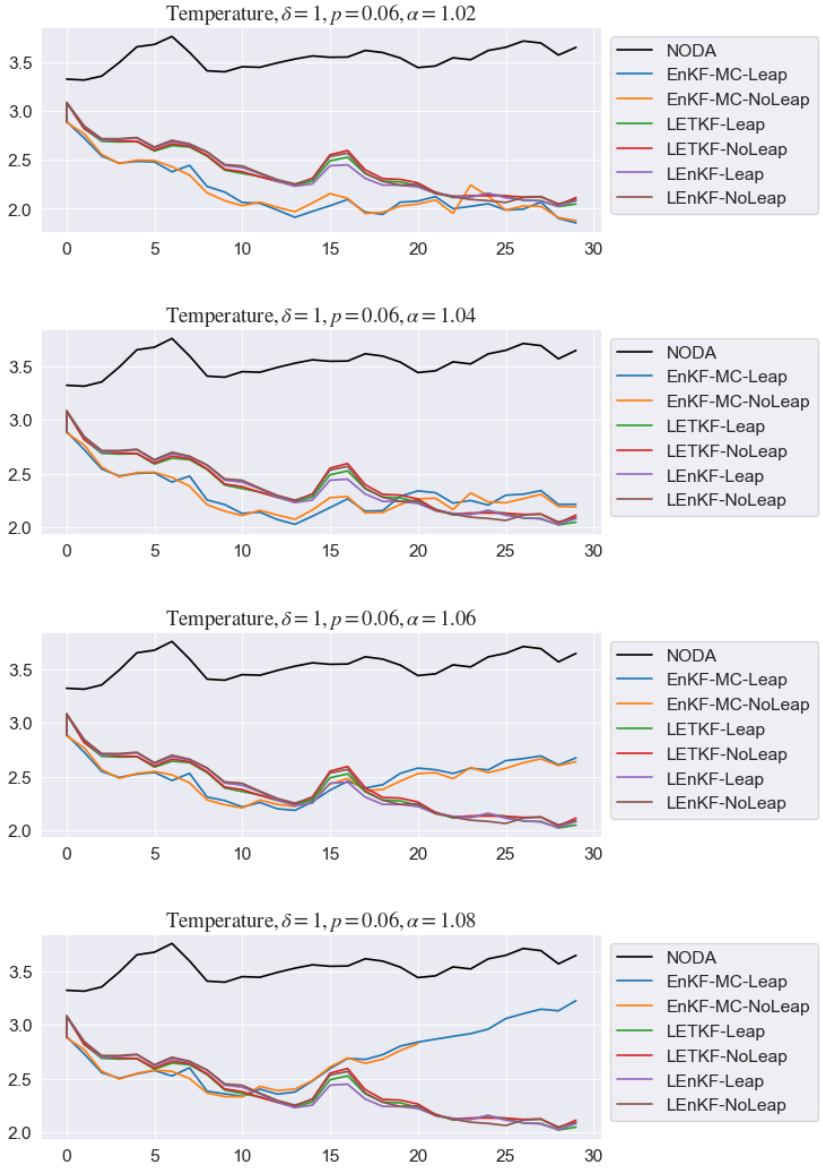


FIGURE 5.46: Time evolution of Temperature for $\delta = 1$ and $p = 6\%$ varying α

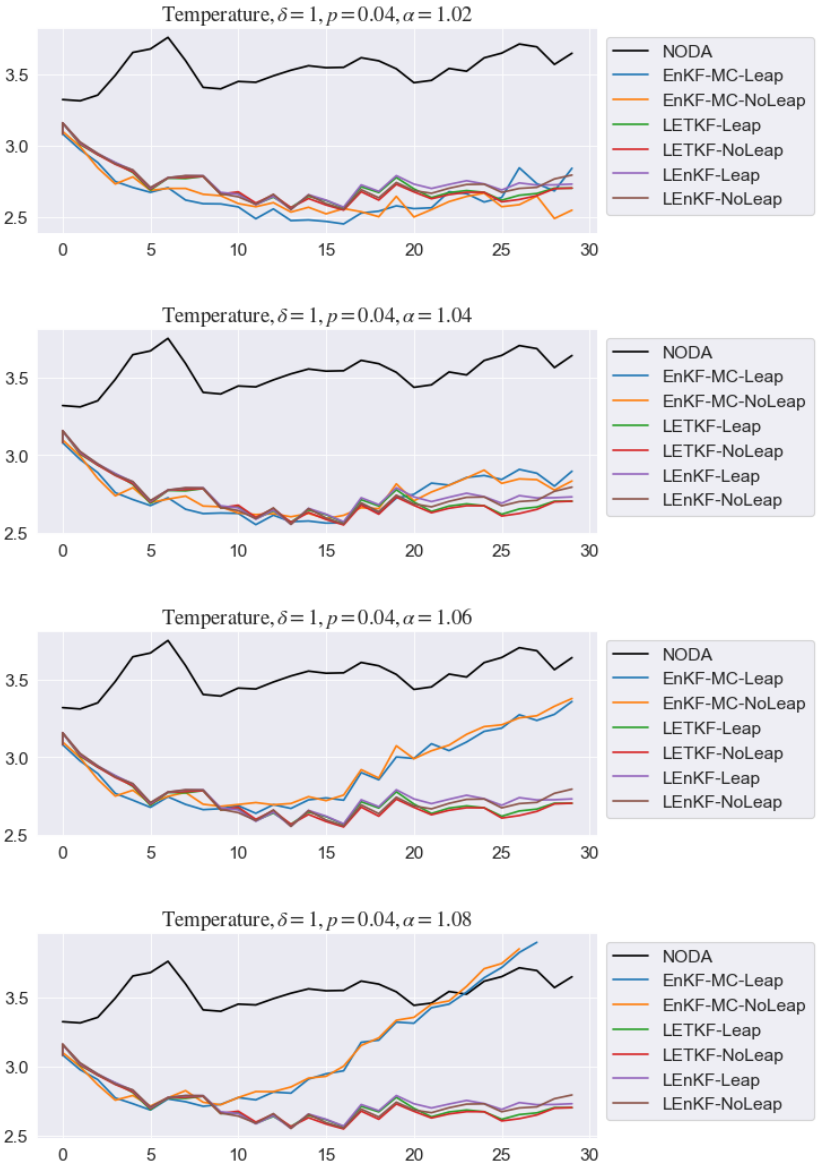


FIGURE 5.47: Time evolution of Temperature for $\delta = 1$ and $p = 4\%$ varying α

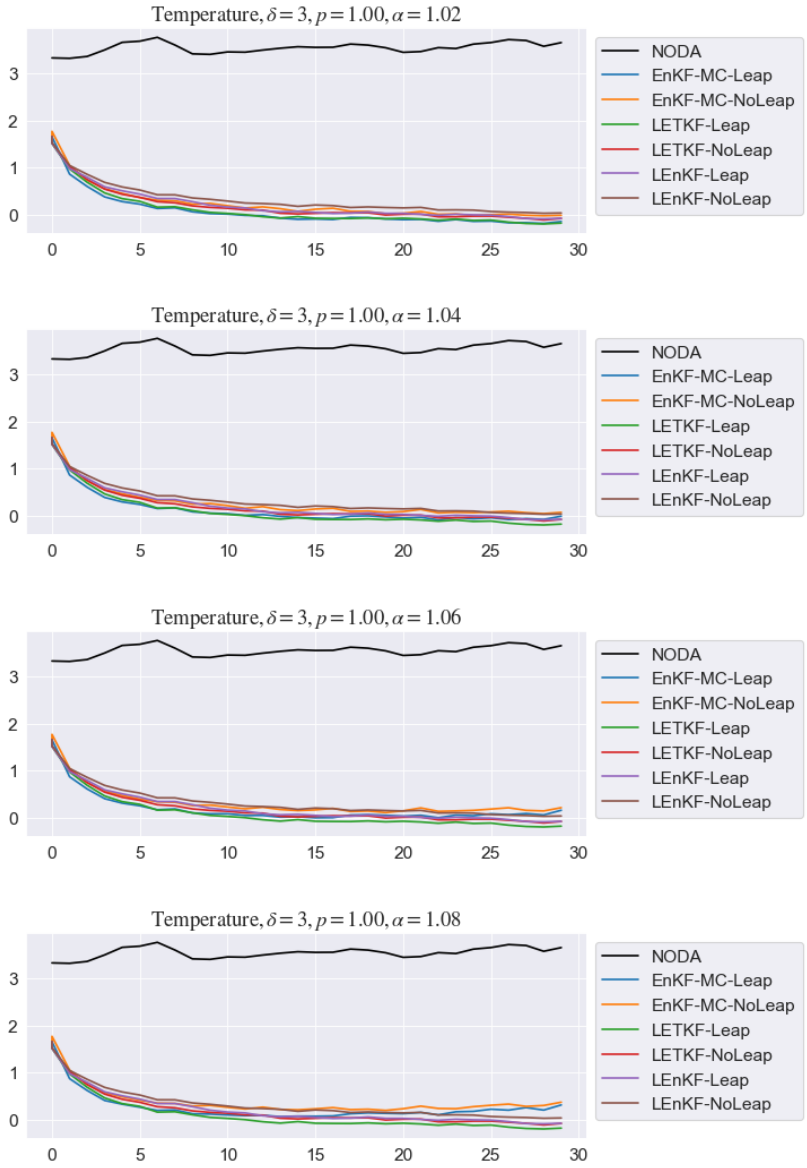


FIGURE 5.48: Time evolution of Temperature for $\delta = 3$ and $p = 100\%$ varying α

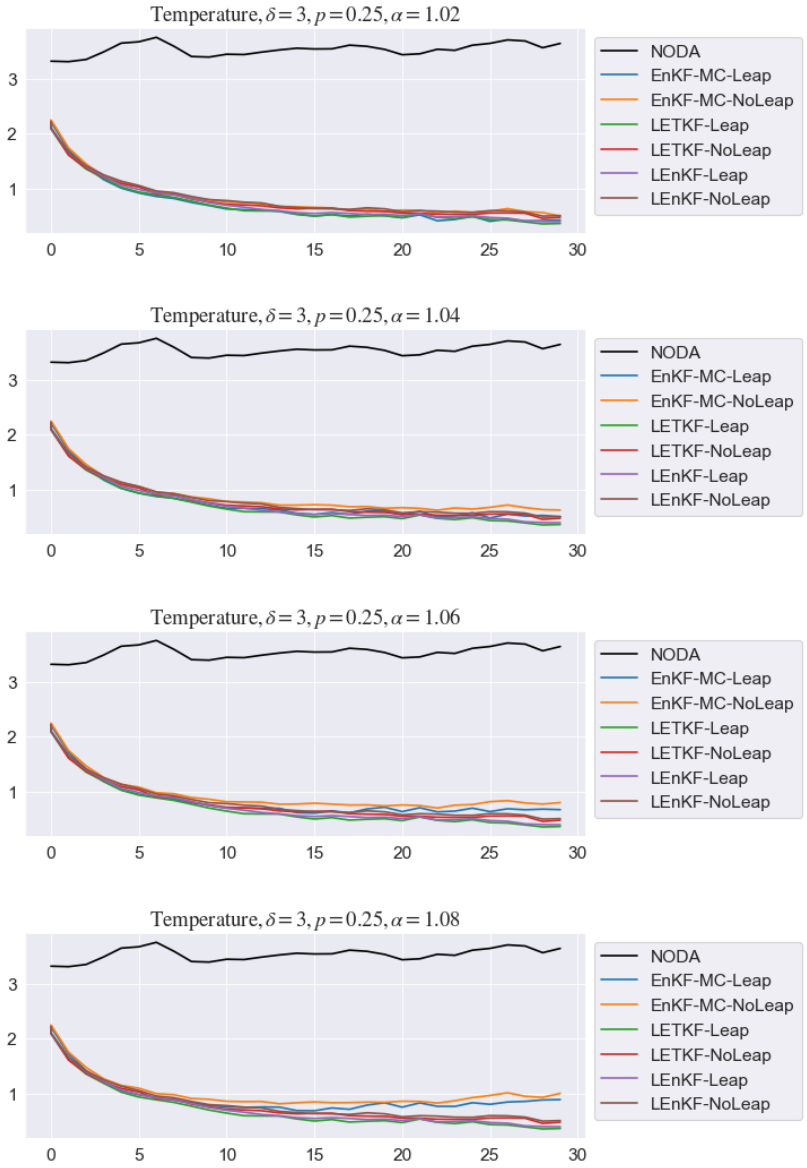


FIGURE 5.49: Time evolution of Temperature for $\delta = 3$ and $p = 25\%$ varying α

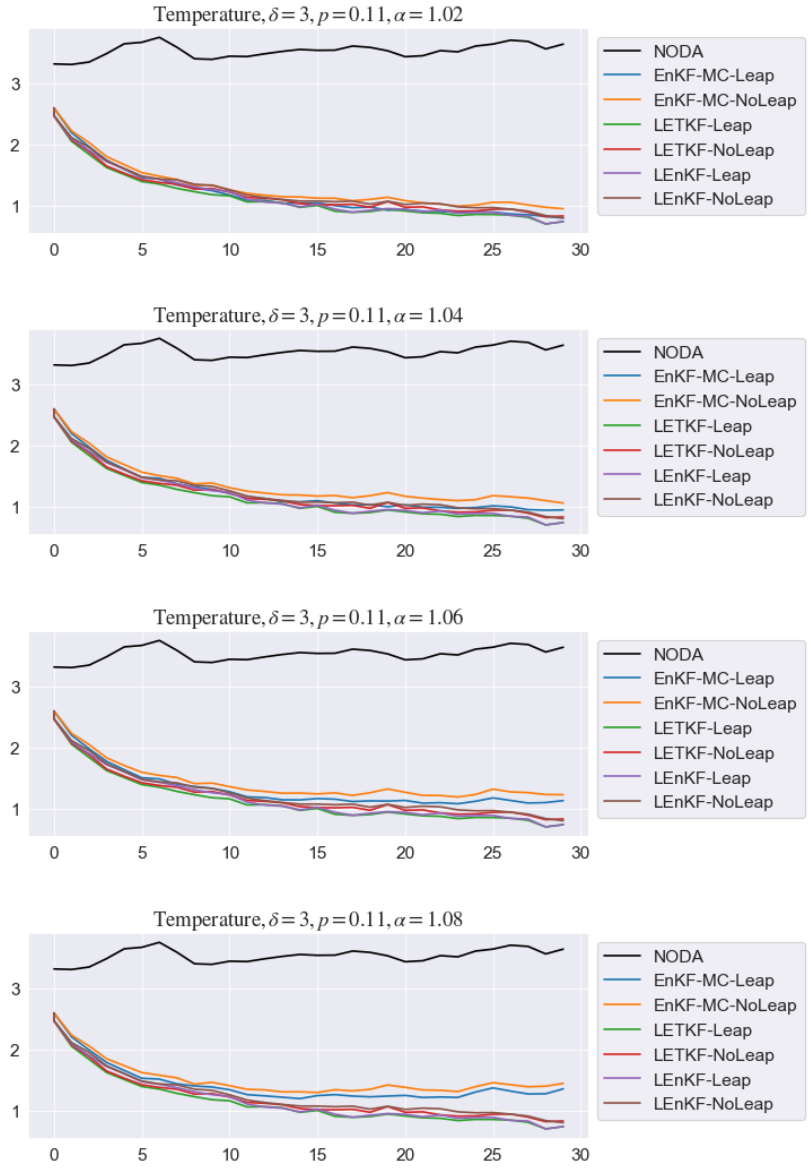


FIGURE 5.50: Time evolution of Temperature for $\delta = 3$ and $p = 11\%$ varying α

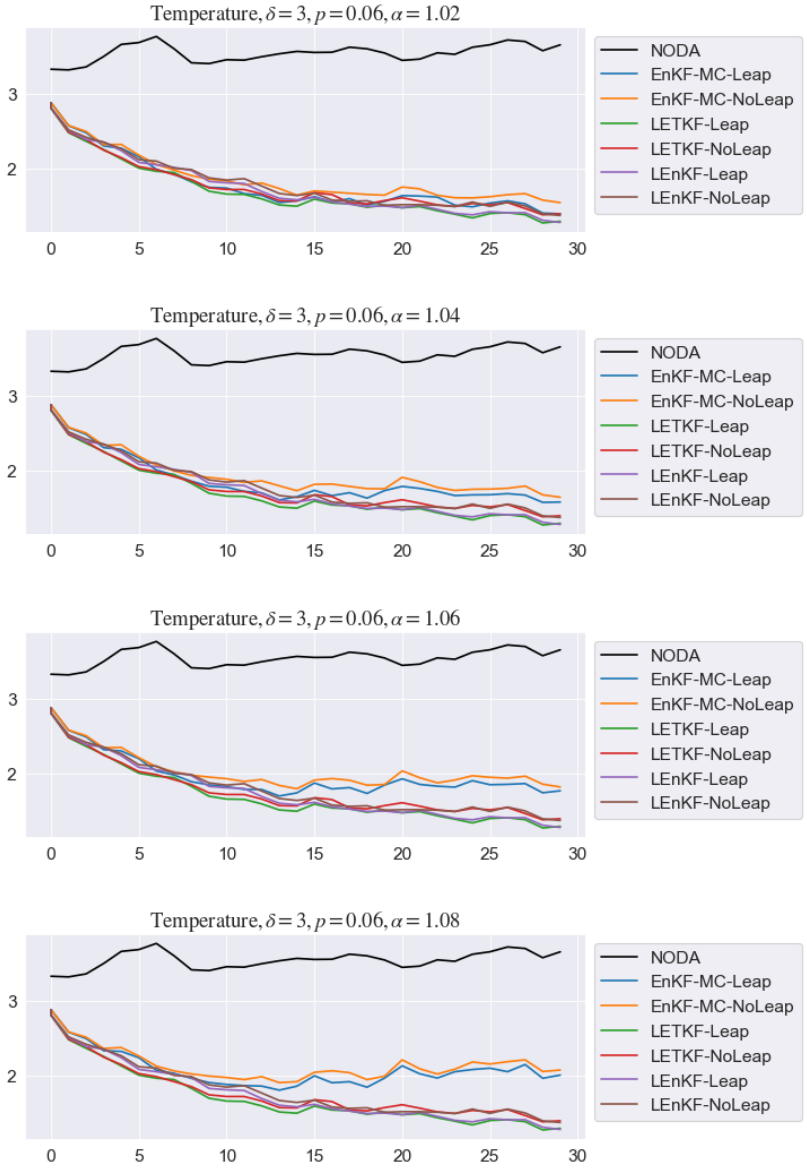


FIGURE 5.51: Time evolution of Temperature for $\delta = 3$ and $p = 6\%$ varying α

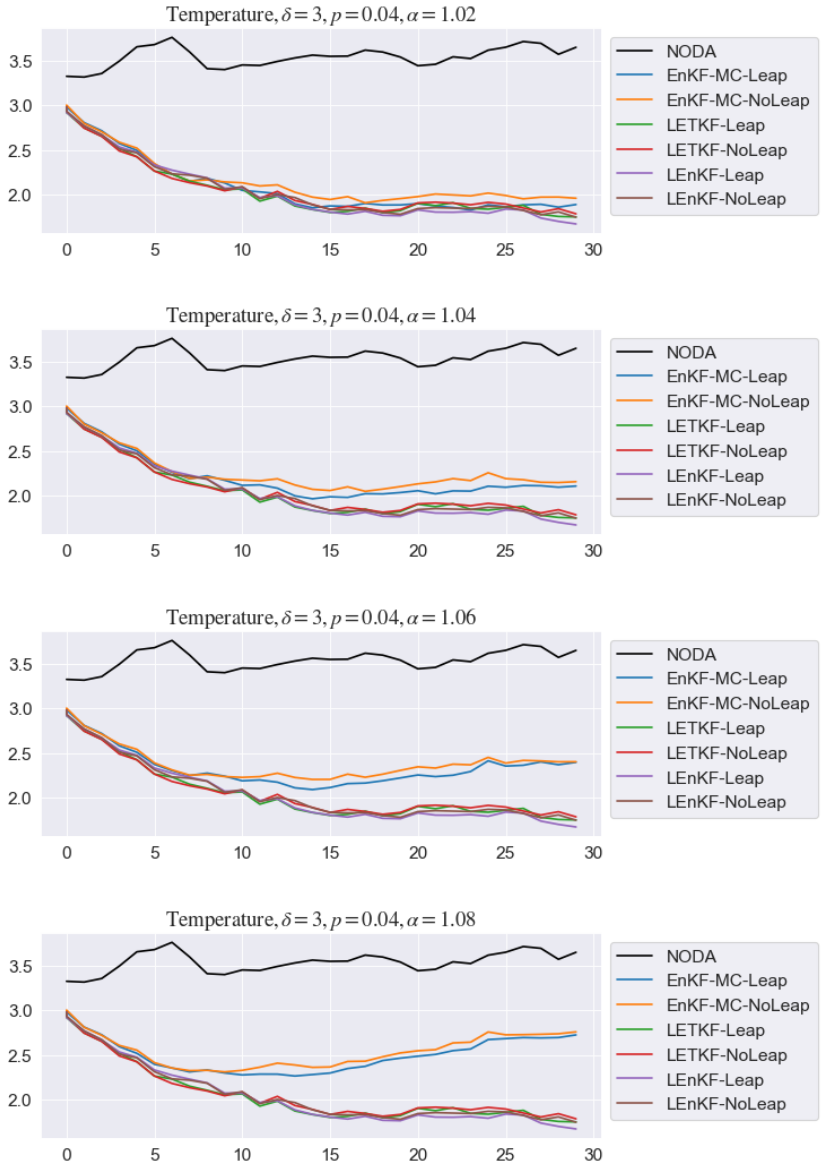


FIGURE 5.52: Time evolution of Temperature for $\delta = 3$ and $p = 4\%$ varying α

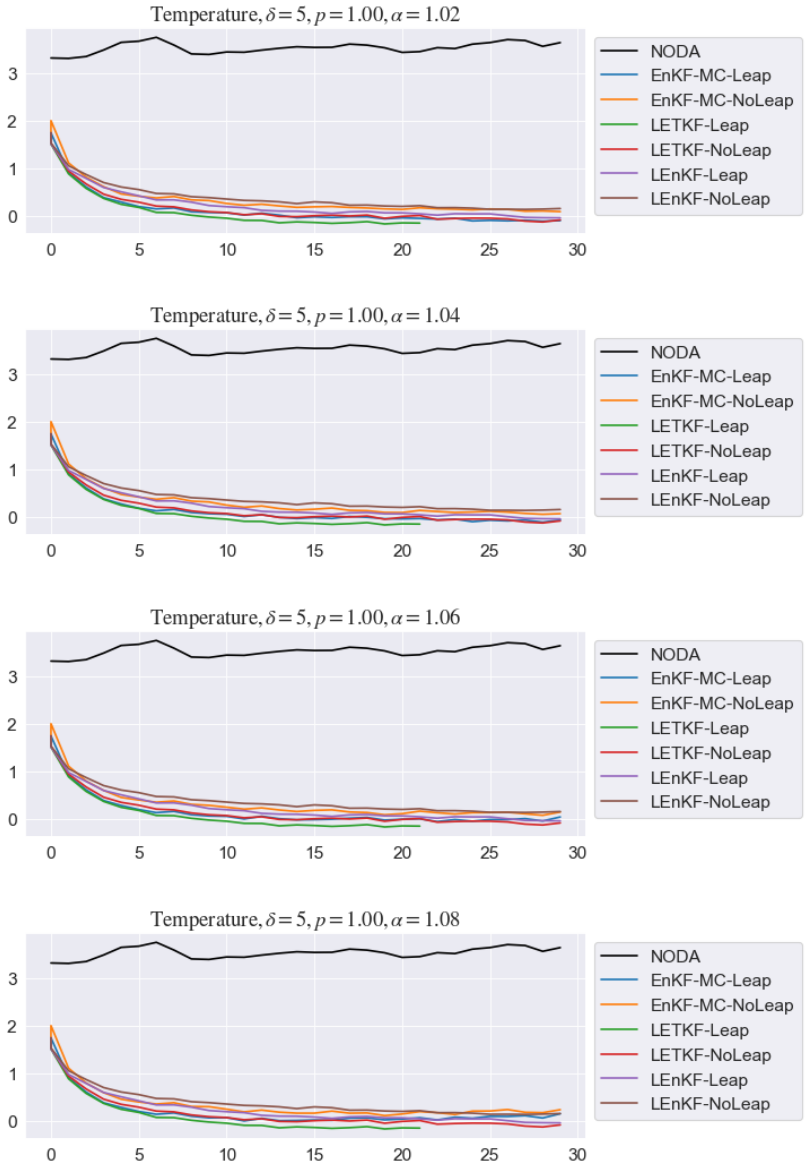


FIGURE 5.53: Time evolution of Temperature for $\delta = 5$ and $p = 100\%$ varying α

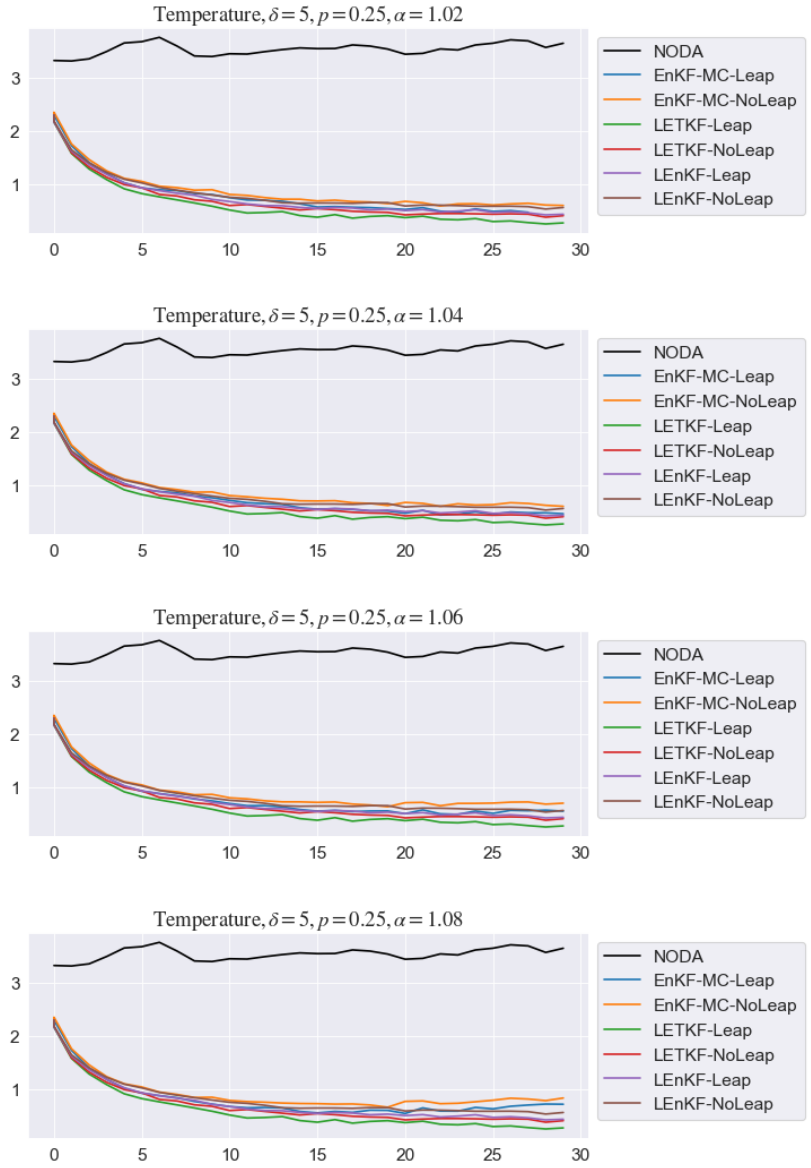


FIGURE 5.54: Time evolution of Temperature for $\delta = 5$ and $p = 25\%$ varying α

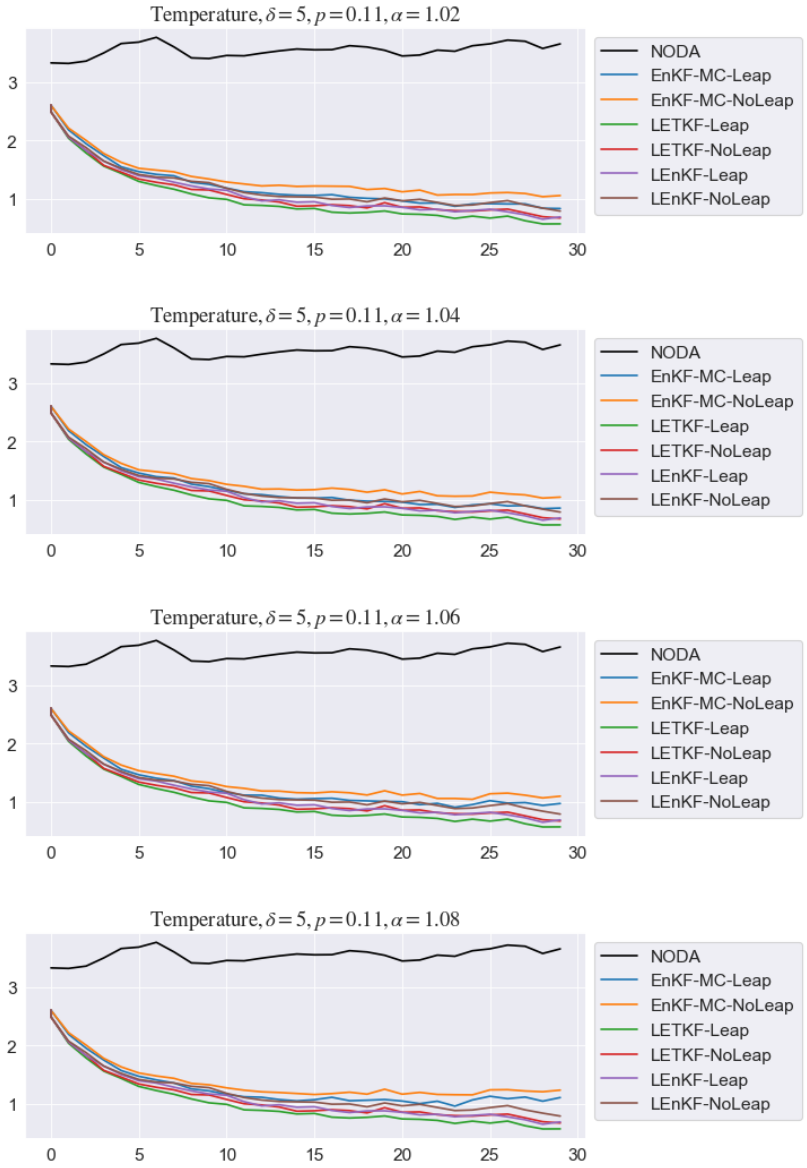


FIGURE 5.55: Time evolution of Temperature for $\delta = 5$ and $p = 11\%$ varying α

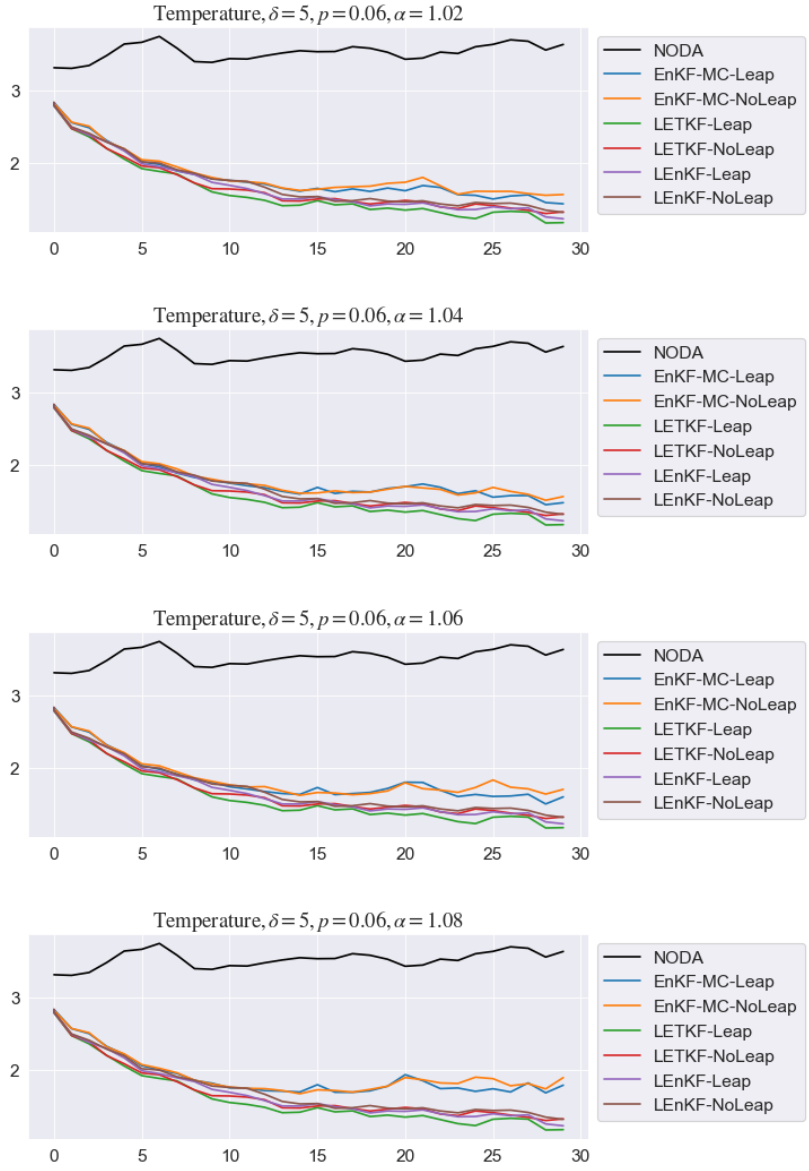


FIGURE 5.56: Time evolution of Temperature for $\delta = 5$ and $p = 6\%$ varying α

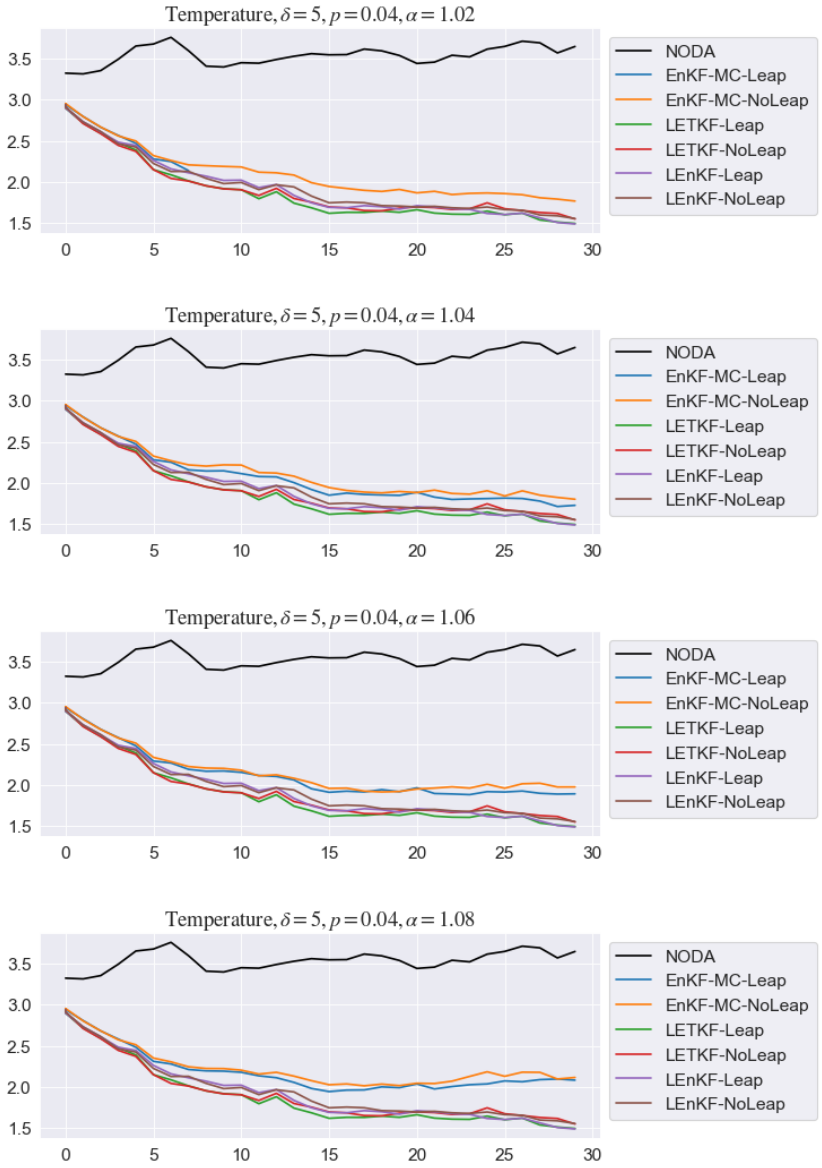


FIGURE 5.57: Time evolution of Temperature for $\delta = 5$ and $p = 4\%$ varying α

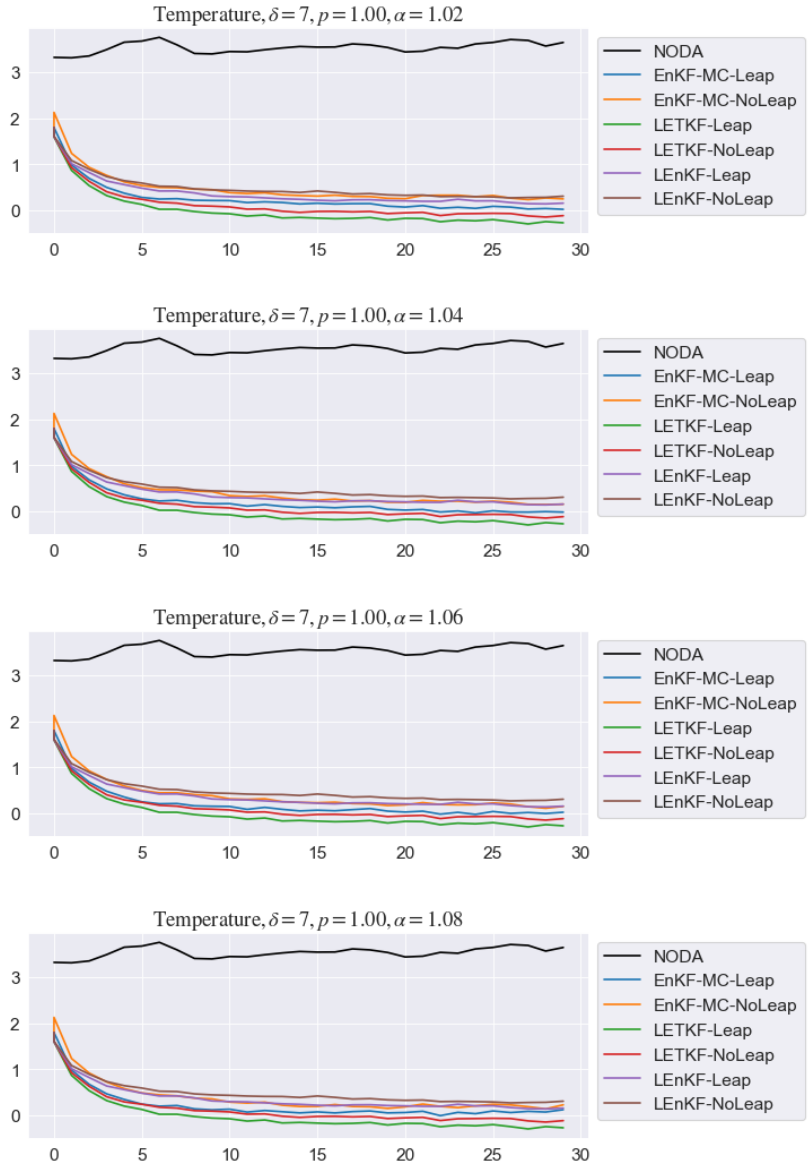


FIGURE 5.58: Time evolution of Temperature for $\delta = 7$ and $p = 100\%$ varying α

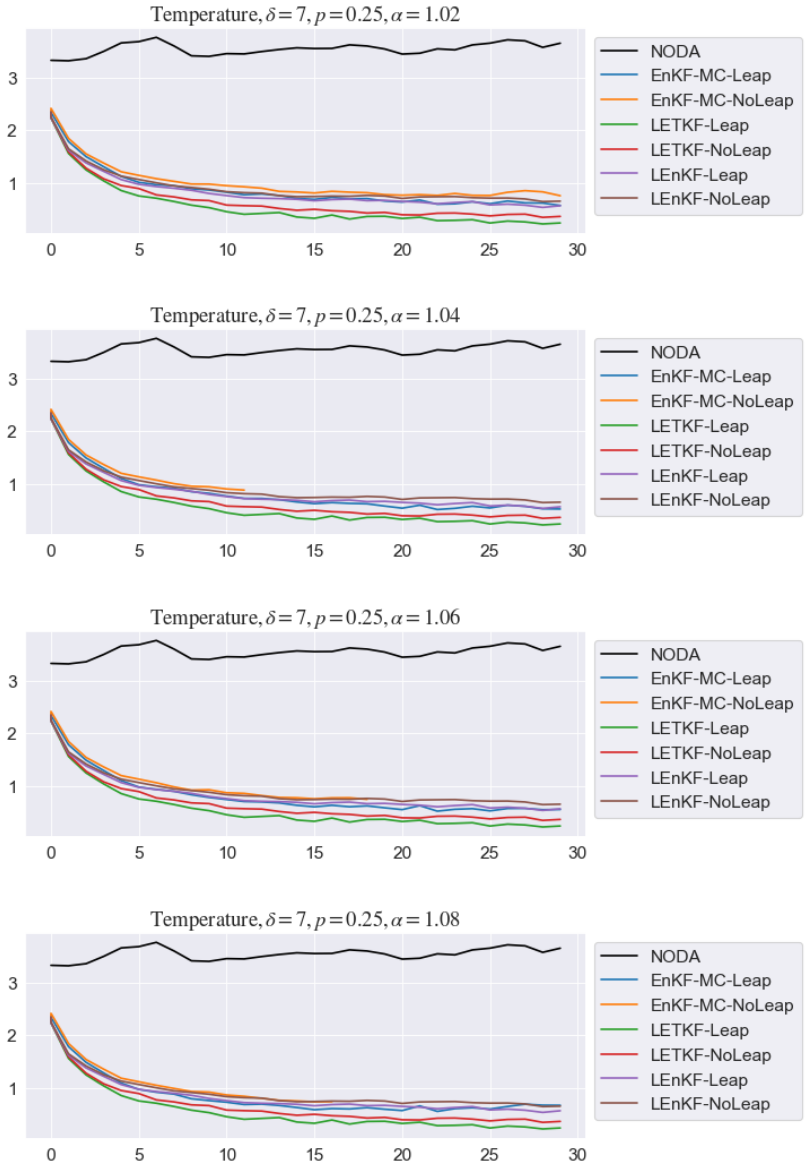


FIGURE 5.59: Time evolution of Temperature for $\delta = 7$ and $p = 25\%$ varying α

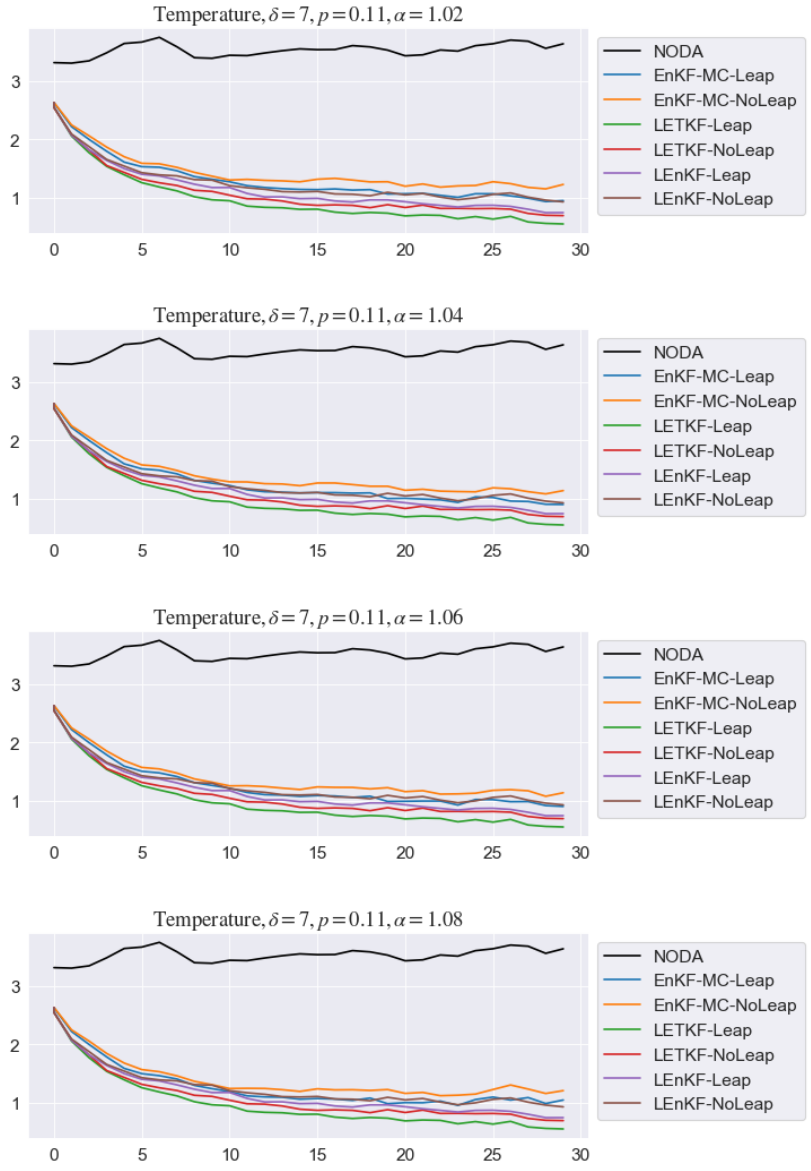


FIGURE 5.60: Time evolution of Temperature for $\delta = 7$ and $p = 11\%$ varying α

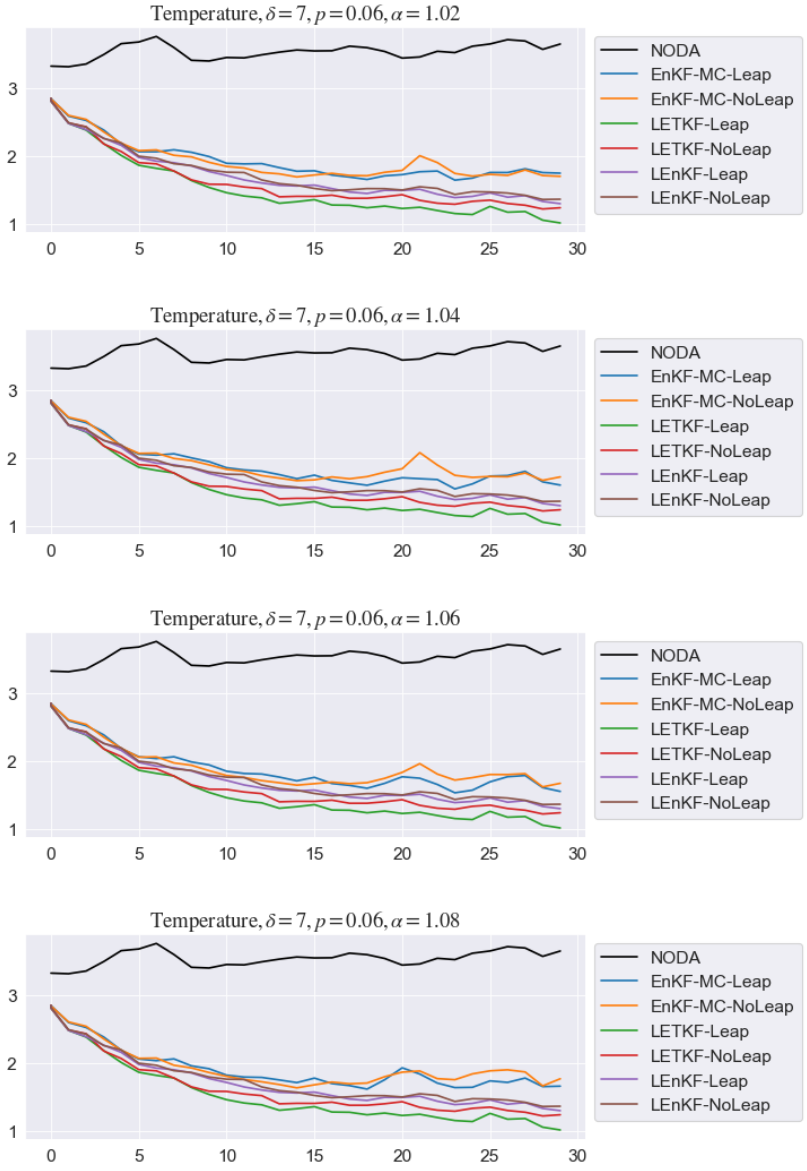


FIGURE 5.61: Time evolution of Temperature for $\delta = 7$ and $p = 6\%$ varying α

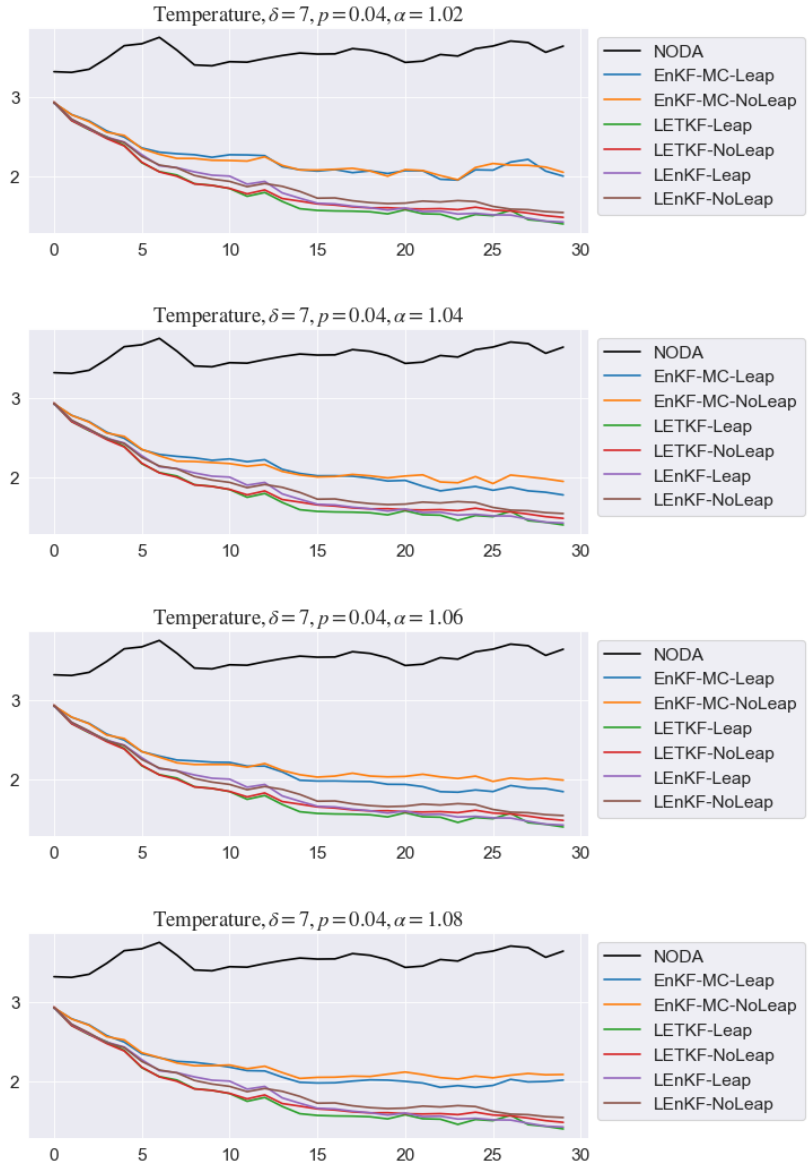


FIGURE 5.62: Time evolution of Temperature for $\delta = 7$ and $p = 4\%$ varying α

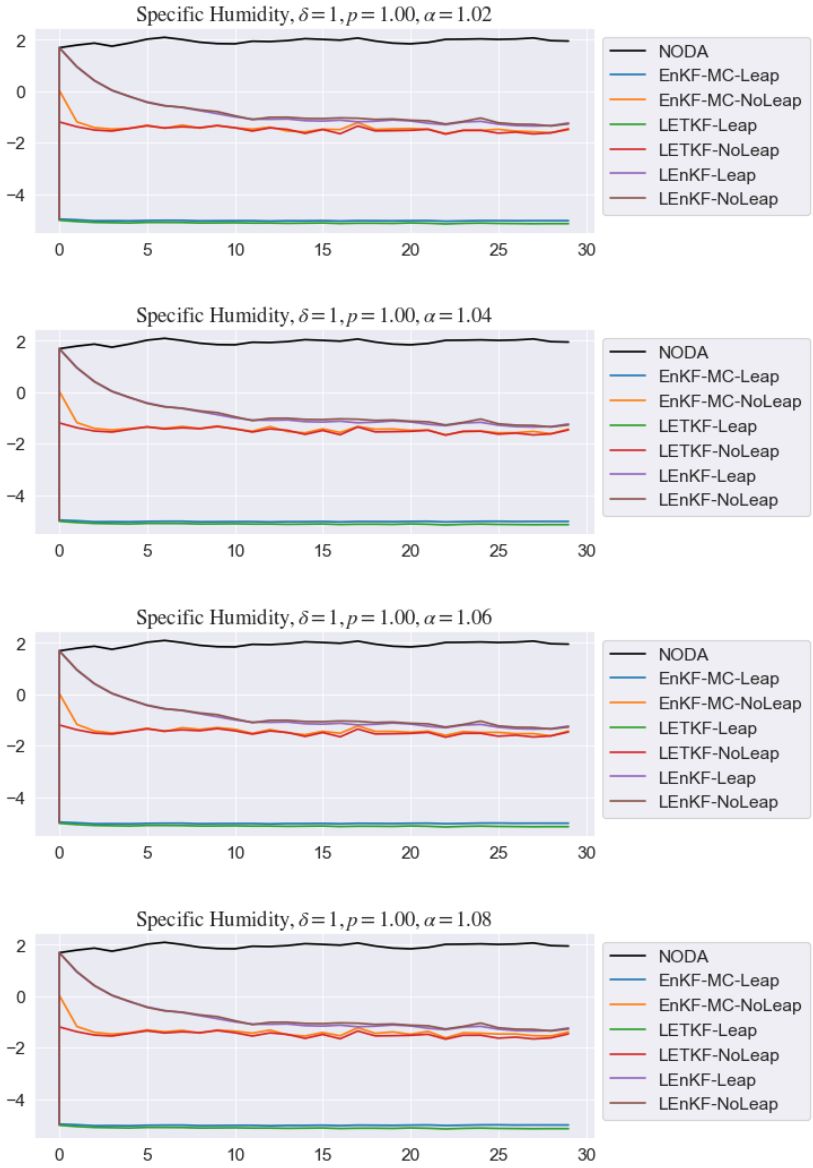


FIGURE 5.63: Time evolution of Specific Humidity for $\delta = 1$ and $p = 100\%$ varying α

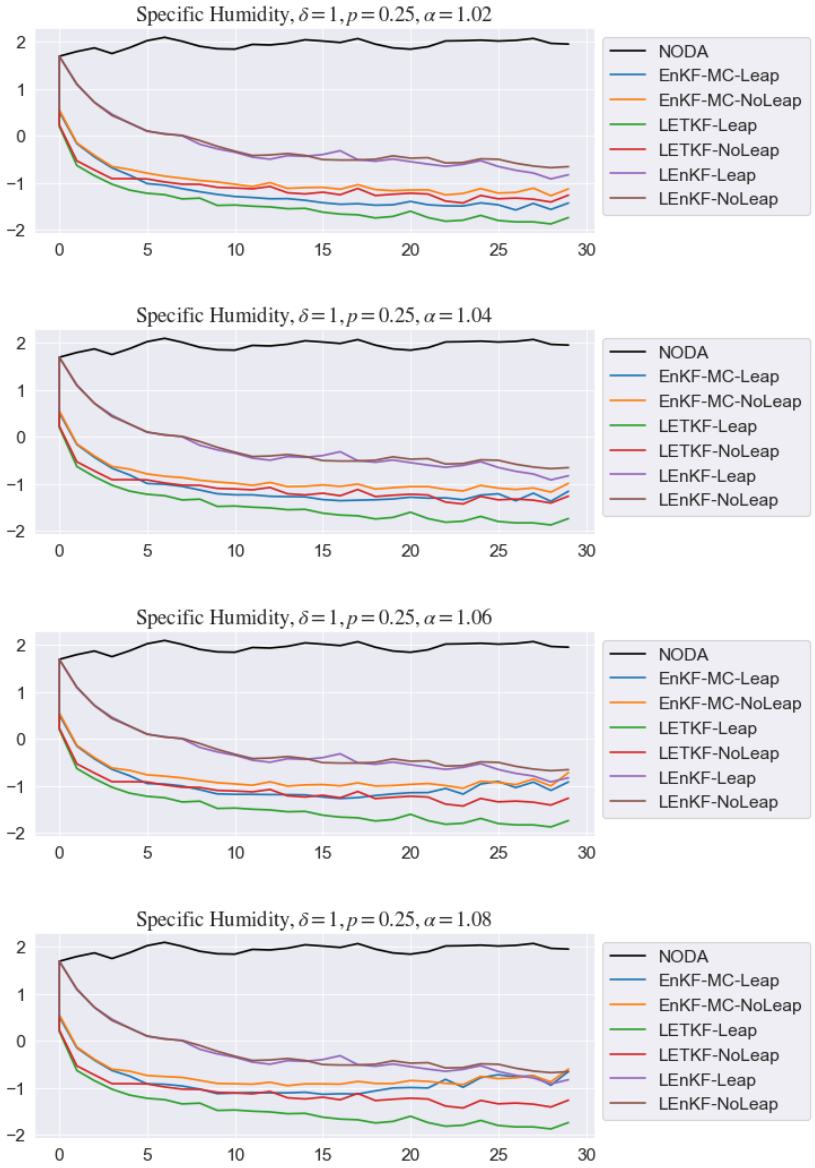


FIGURE 5.64: Time evolution of Specific Humidity for $\delta = 1$ and $p = 25\%$ varying α

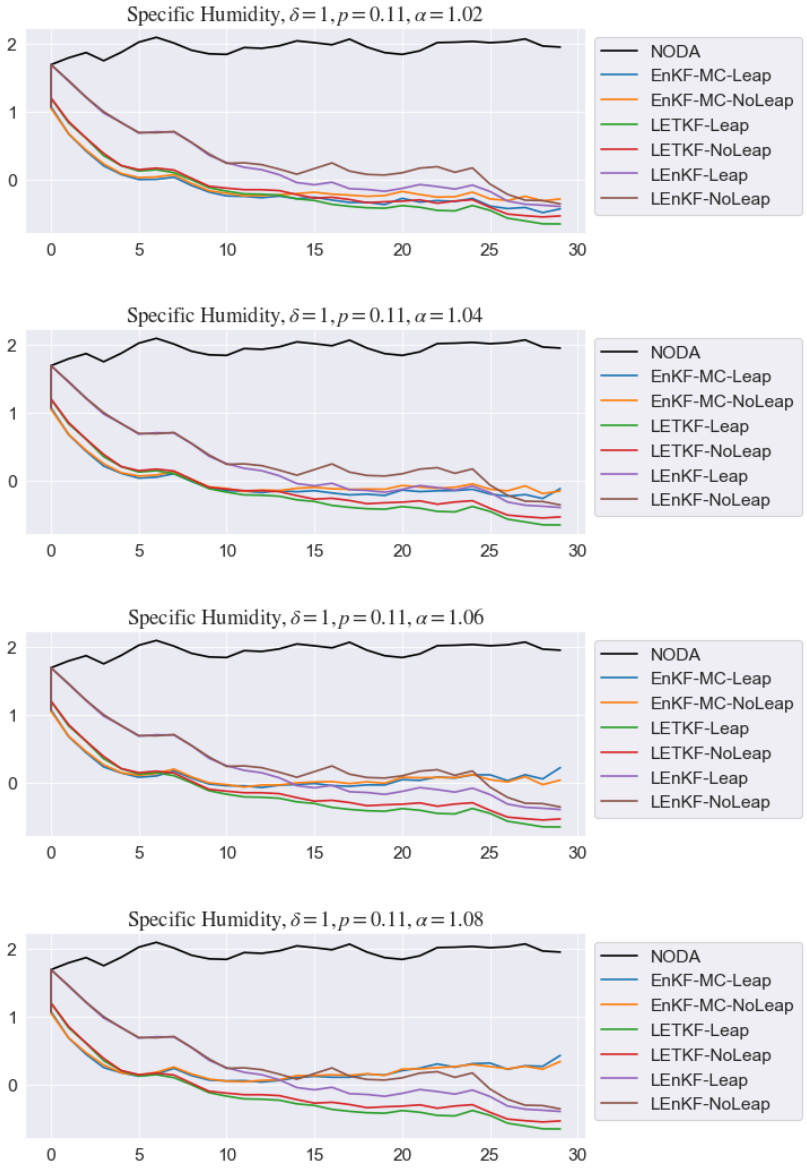


FIGURE 5.65: Time evolution of Specific Humidity for $\delta = 1$ and $p = 11\%$ varying α

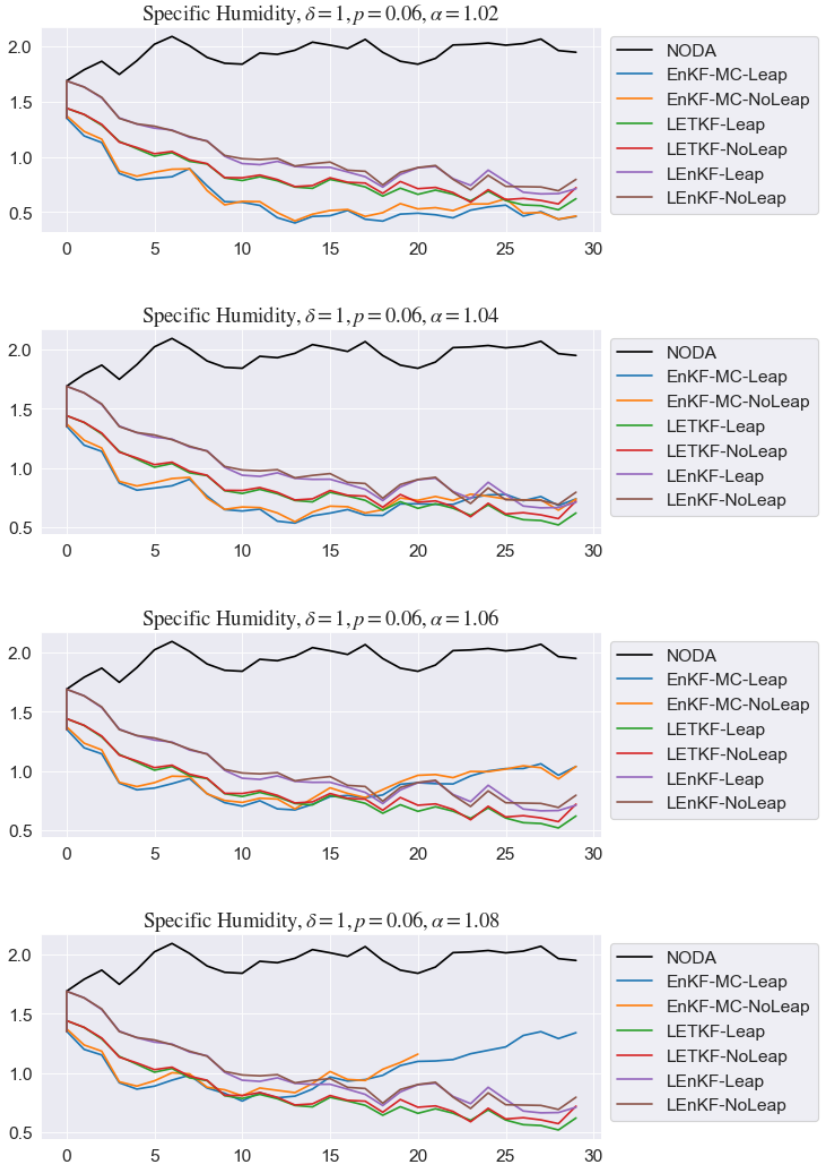


FIGURE 5.66: Time evolution of Specific Humidity for $\delta = 1$ and $p = 6\%$ varying α

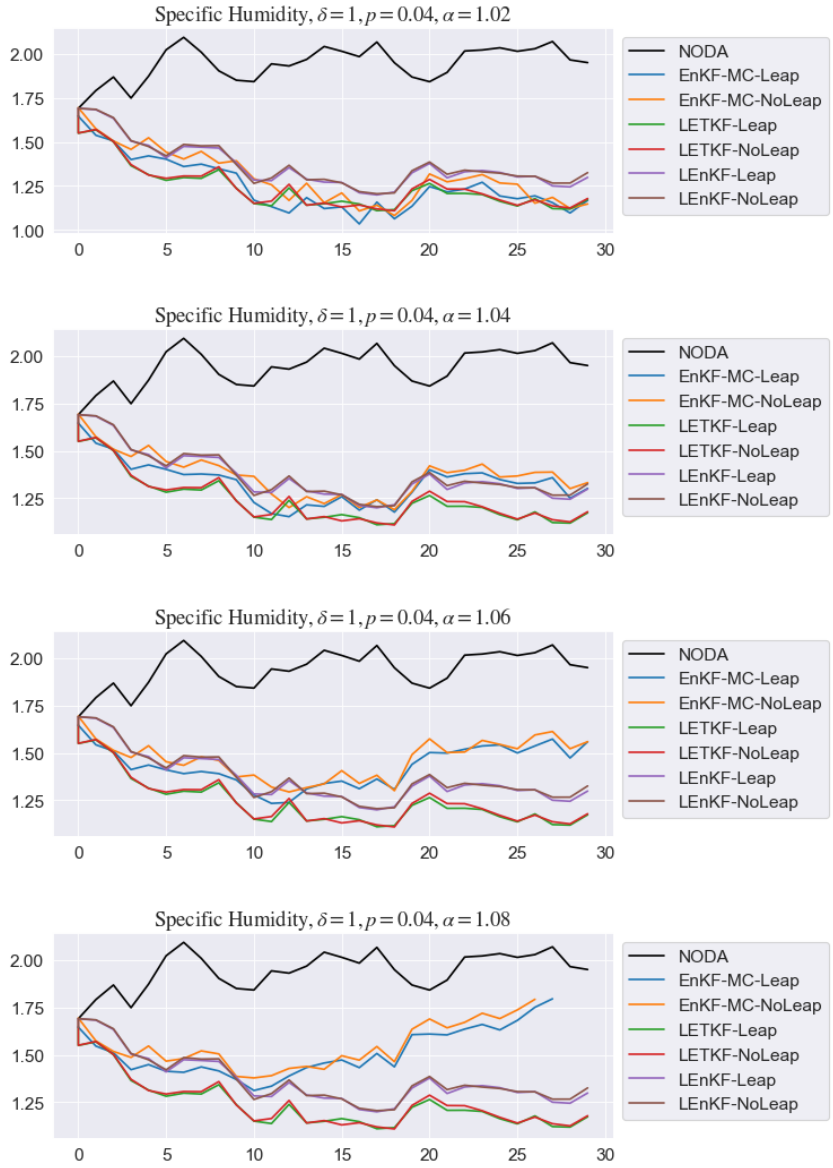


FIGURE 5.67: Time evolution of Specific Humidity for $\delta = 1$ and $p = 4\%$ varying α

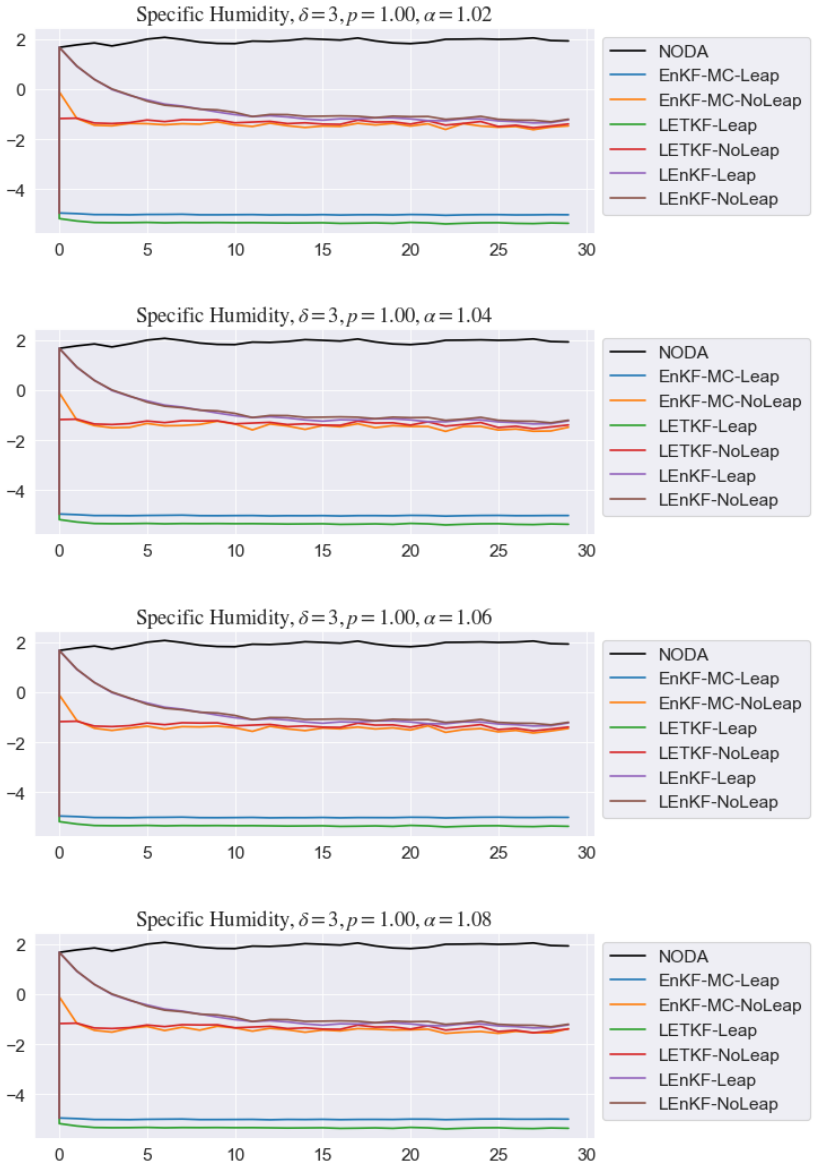


FIGURE 5.68: Time evolution of Specific Humidity for $\delta = 3$ and $p = 100\%$ varying α

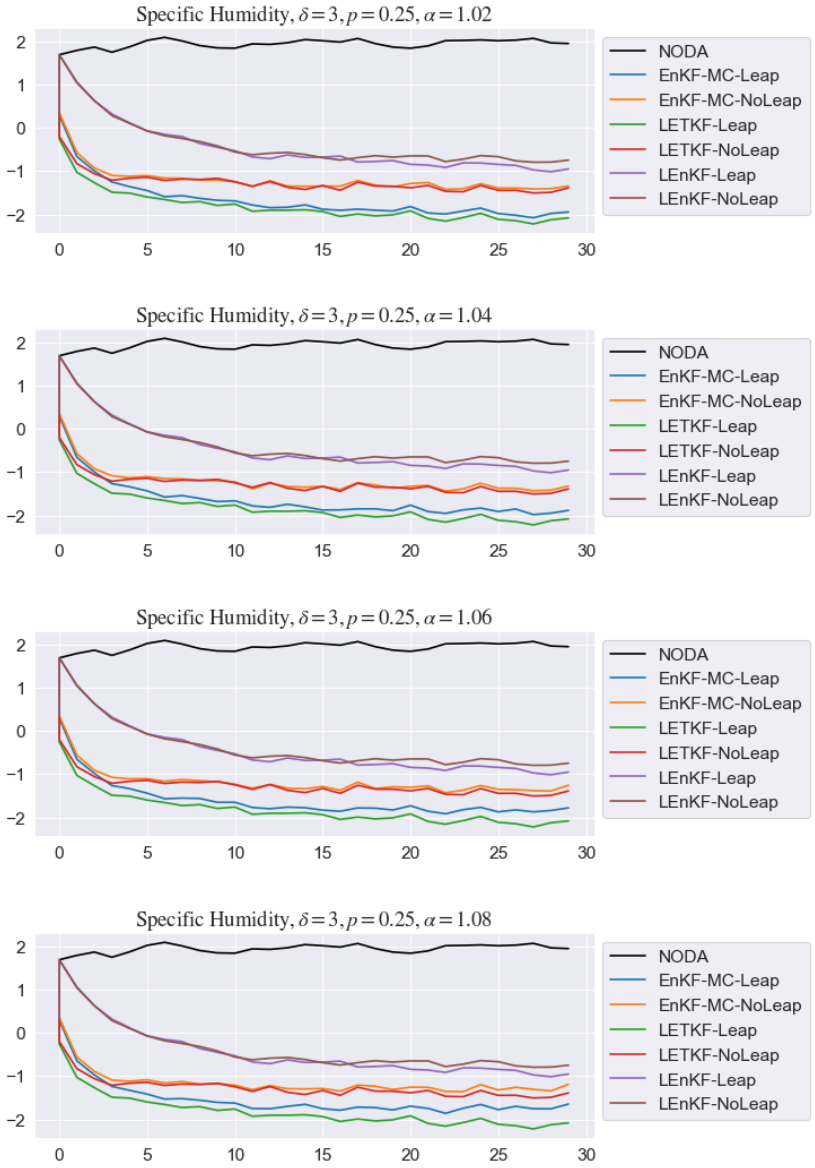


FIGURE 5.69: Time evolution of Specific Humidity for $\delta = 3$ and $p = 25\%$ varying α

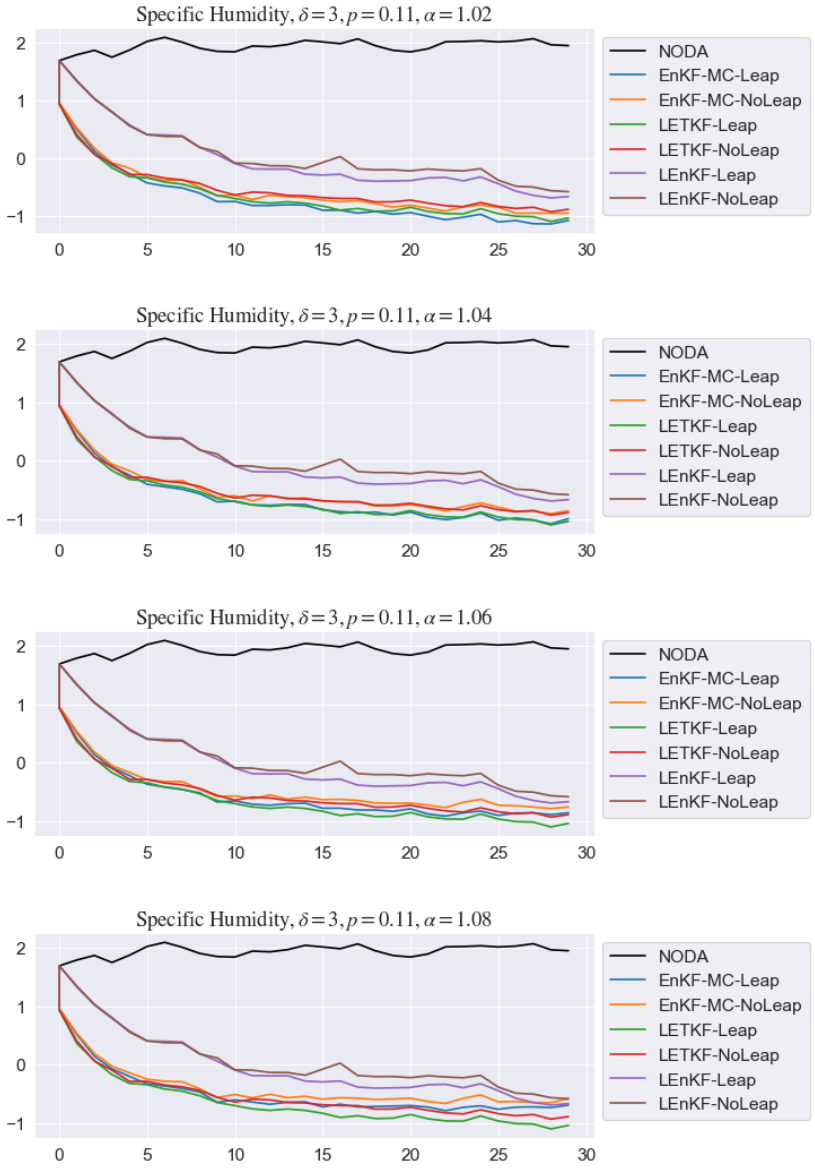


FIGURE 5.70: Time evolution of Specific Humidity for $\delta = 3$ and $p = 11\%$ varying α

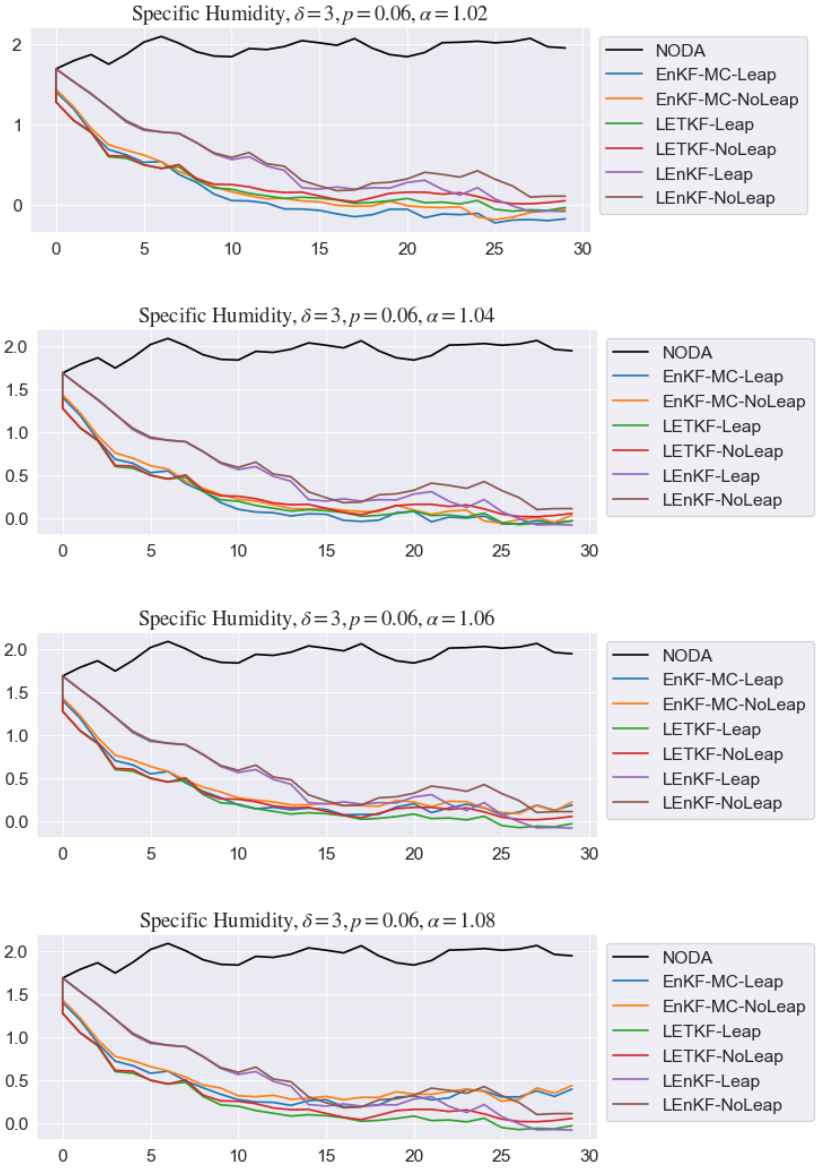


FIGURE 5.71: Time evolution of Specific Humidity for $\delta = 3$ and $p = 6\%$ varying α

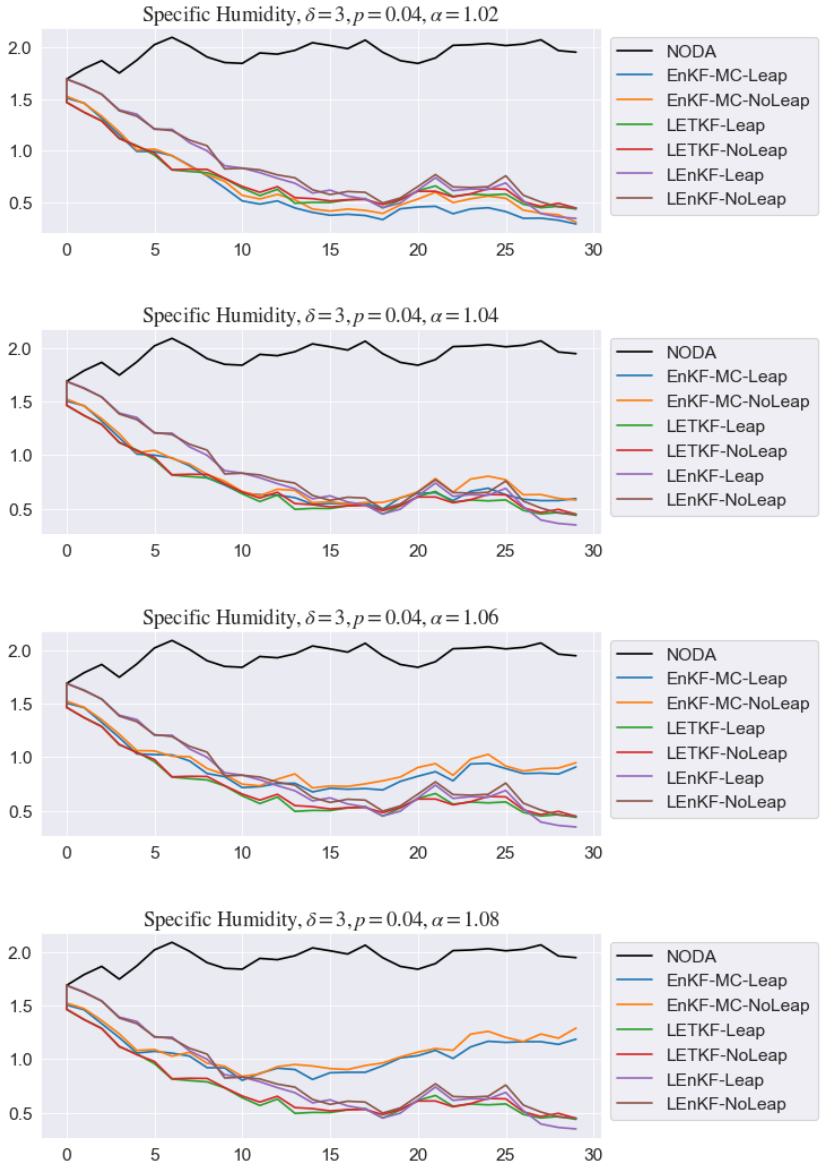


FIGURE 5.72: Time evolution of Specific Humidity for $\delta = 3$ and $p = 4\%$ varying α

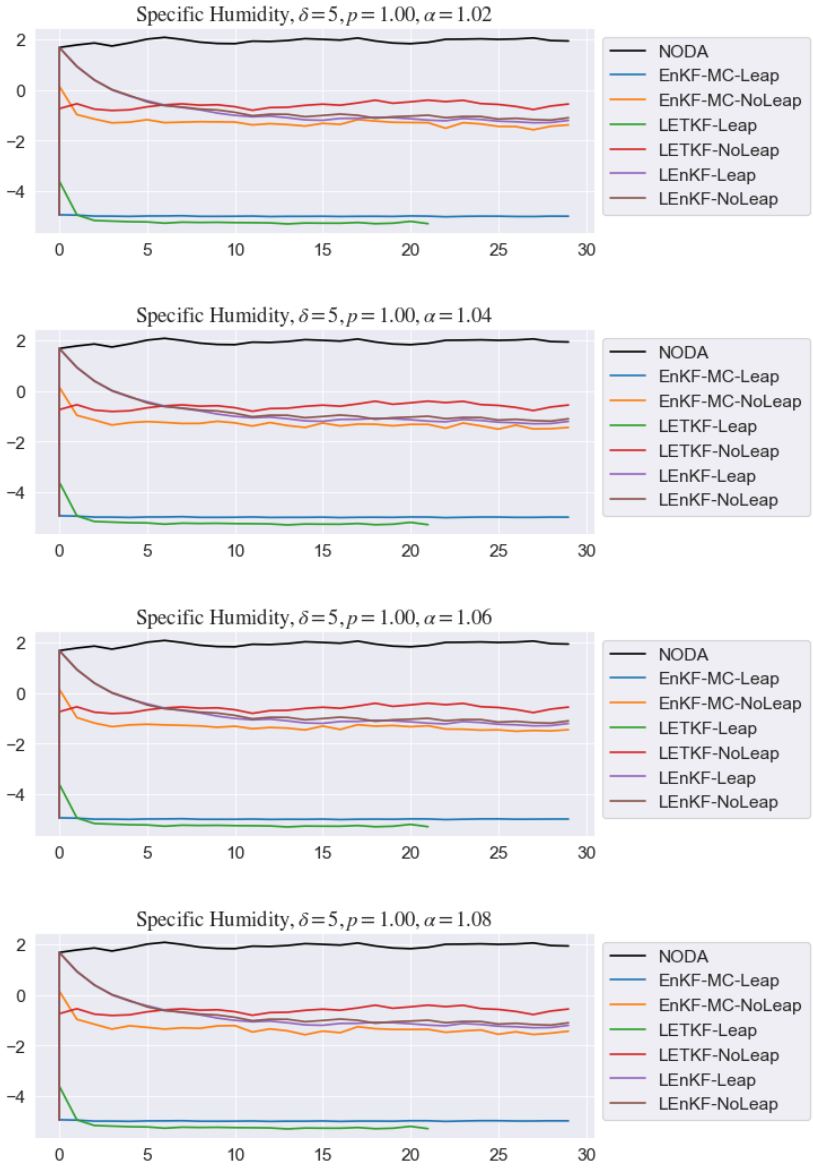


FIGURE 5.73: Time evolution of Specific Humidity for $\delta = 5$ and $p = 100\%$ varying α

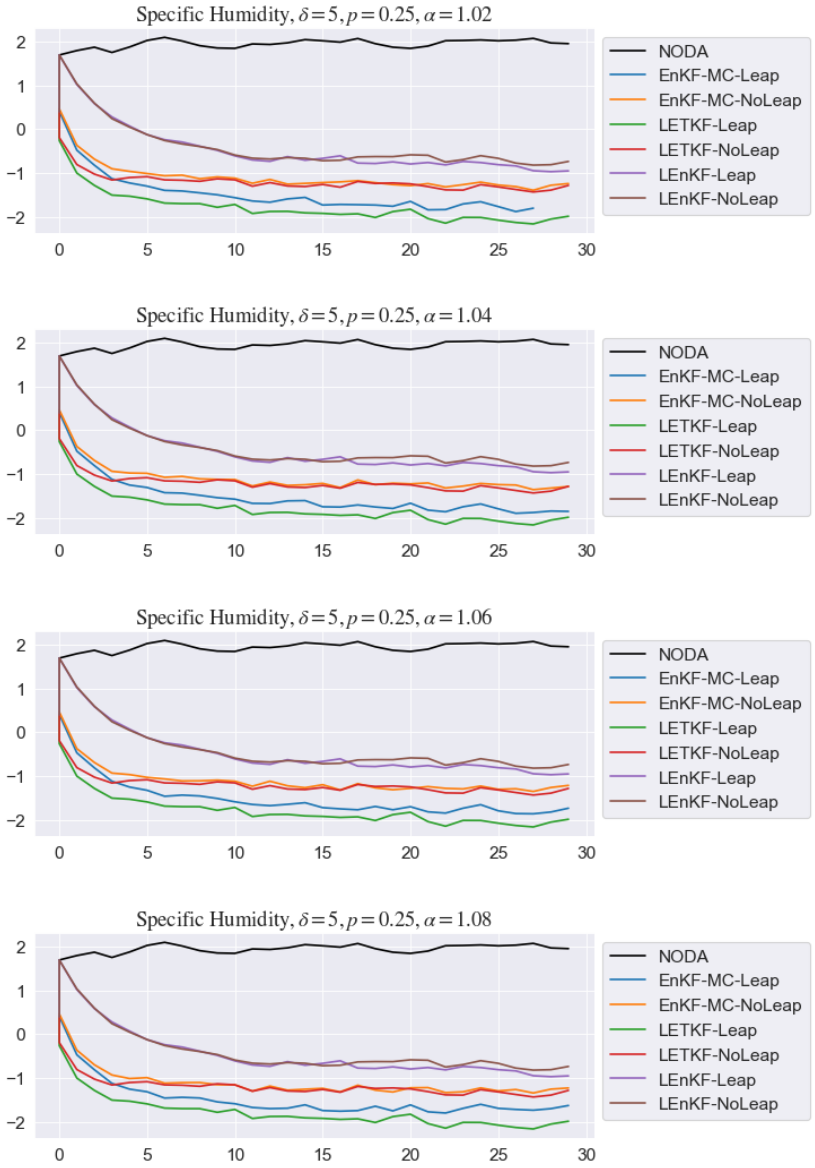


FIGURE 5.74: Time evolution of Specific Humidity for $\delta = 5$ and $p = 25\%$ varying α

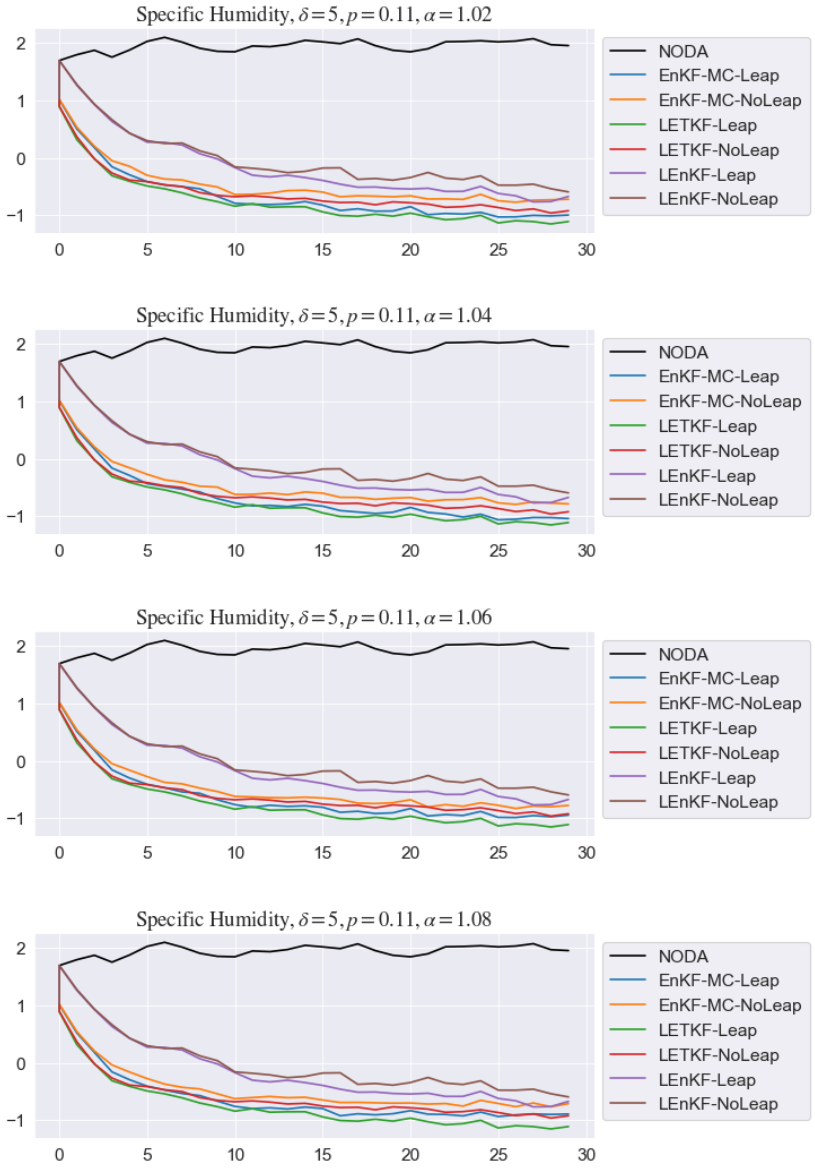


FIGURE 5.75: Time evolution of Specific Humidity for $\delta = 5$ and $p = 11\%$ varying α

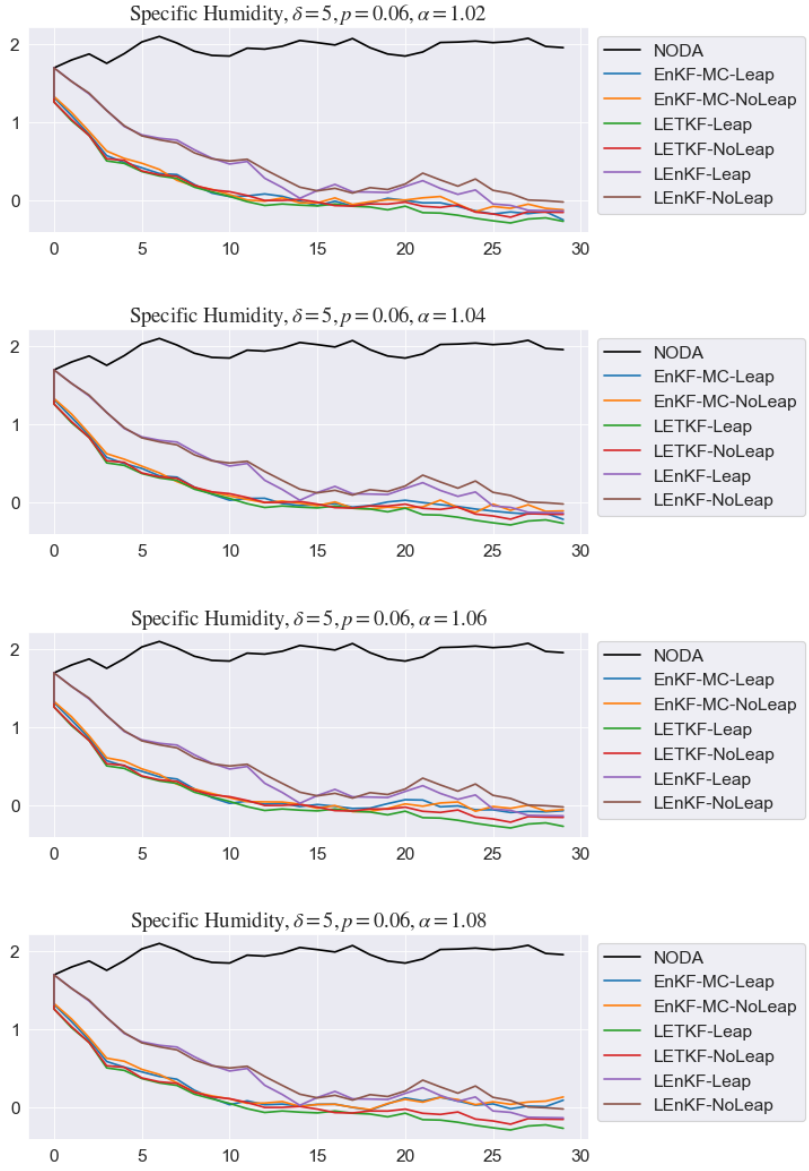


FIGURE 5.76: Time evolution of Specific Humidity for $\delta = 5$ and $p = 6\%$ varying α

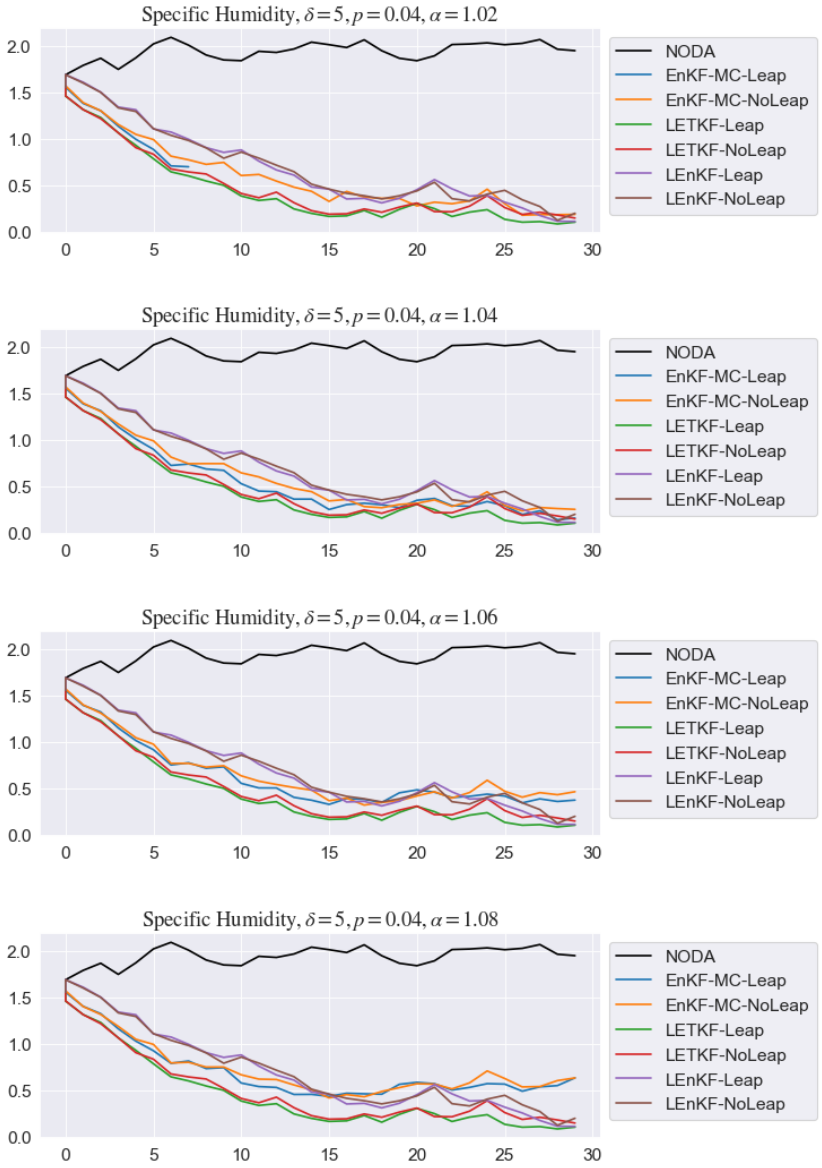


FIGURE 5.77: Time evolution of Specific Humidity for $\delta = 5$ and $p = 4\%$ varying α

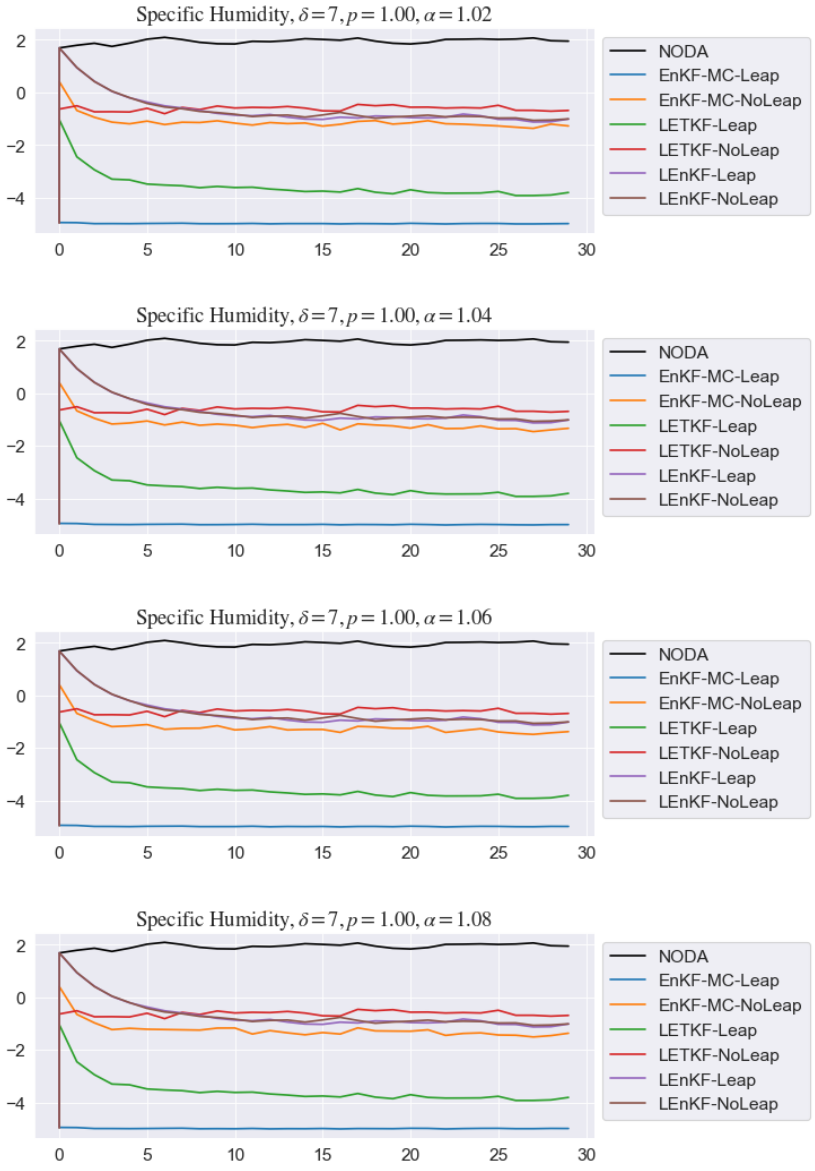


FIGURE 5.78: Time evolution of Specific Humidity for $\delta = 7$ and $p = 100\%$ varying α

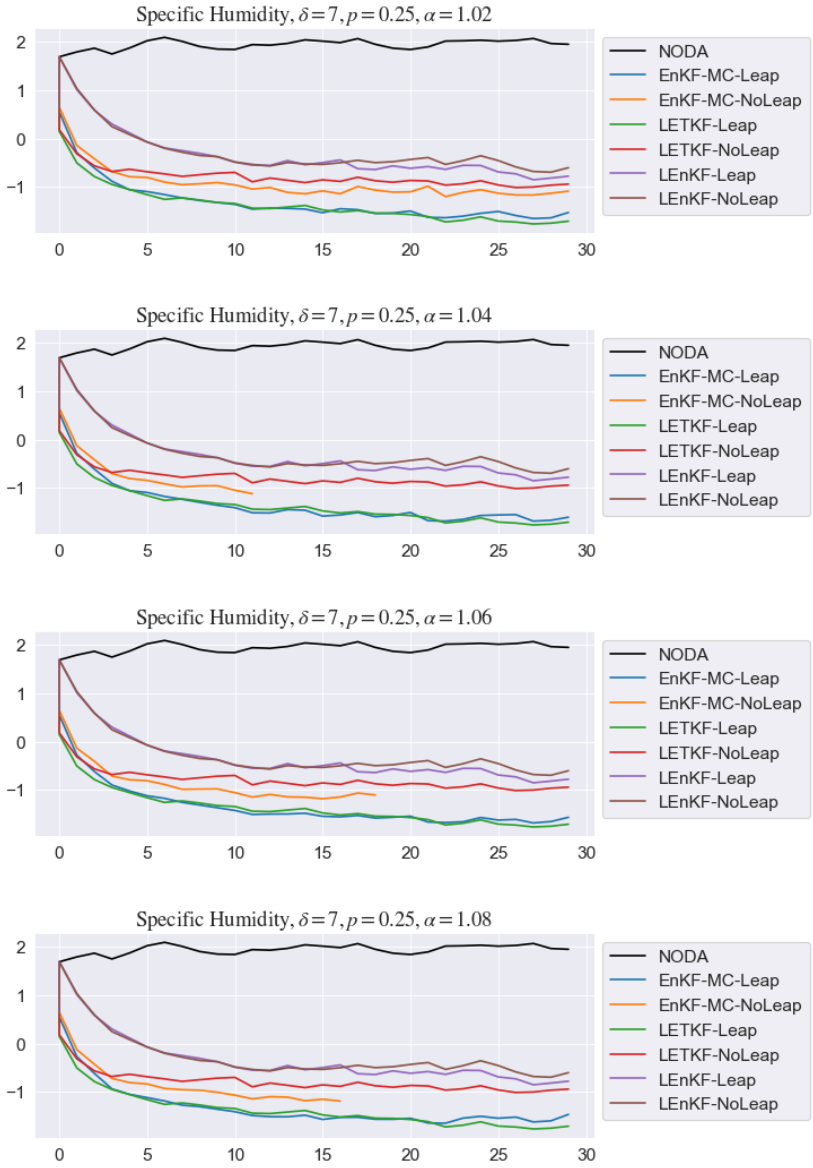


FIGURE 5.79: Time evolution of Specific Humidity for $\delta = 7$ and $p = 25\%$ varying α

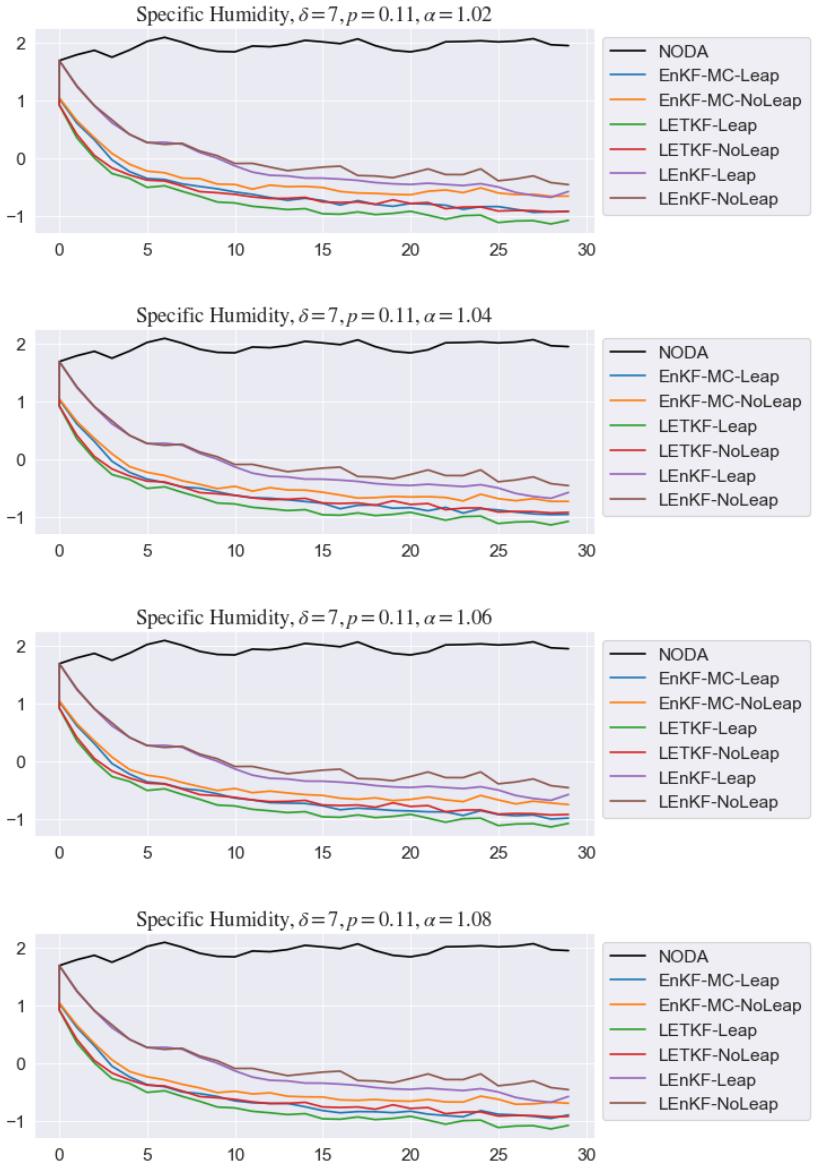


FIGURE 5.80: Time evolution of Specific Humidity for $\delta = 7$ and $p = 11\%$ varying α

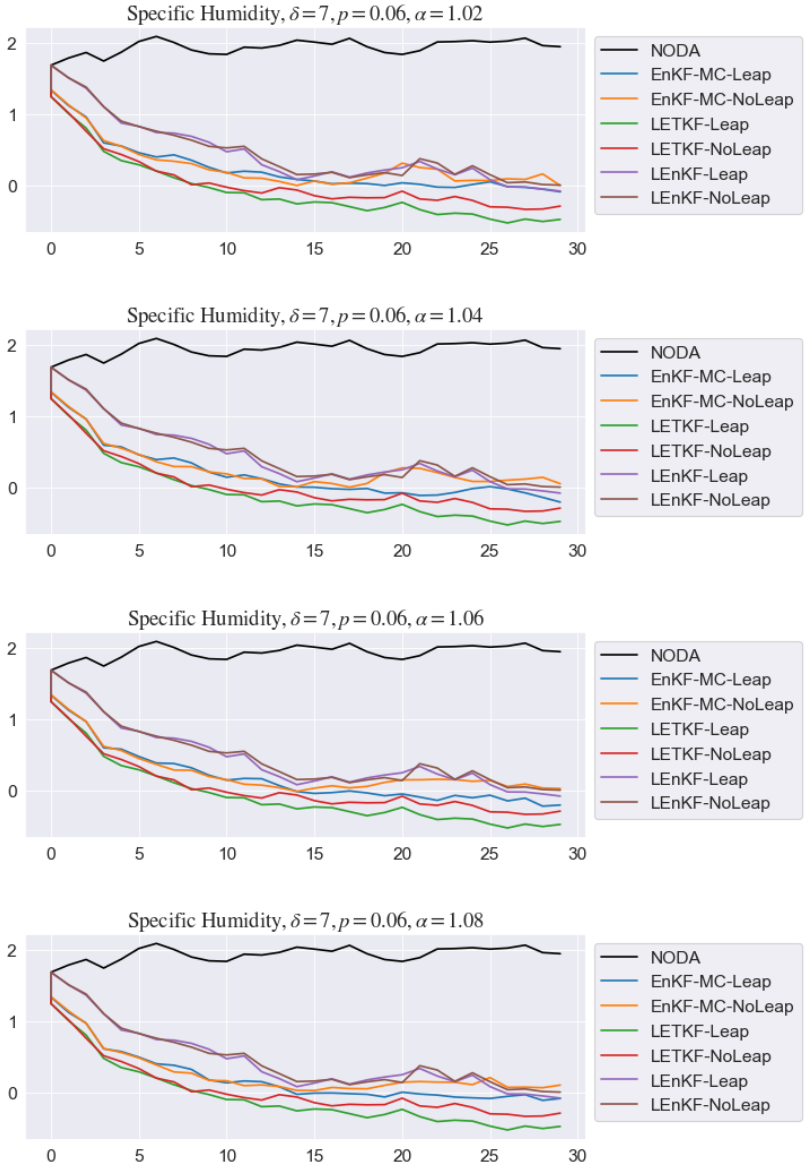


FIGURE 5.81: Time evolution of Specific Humidity for $\delta = 7$ and $p = 6\%$ varying α

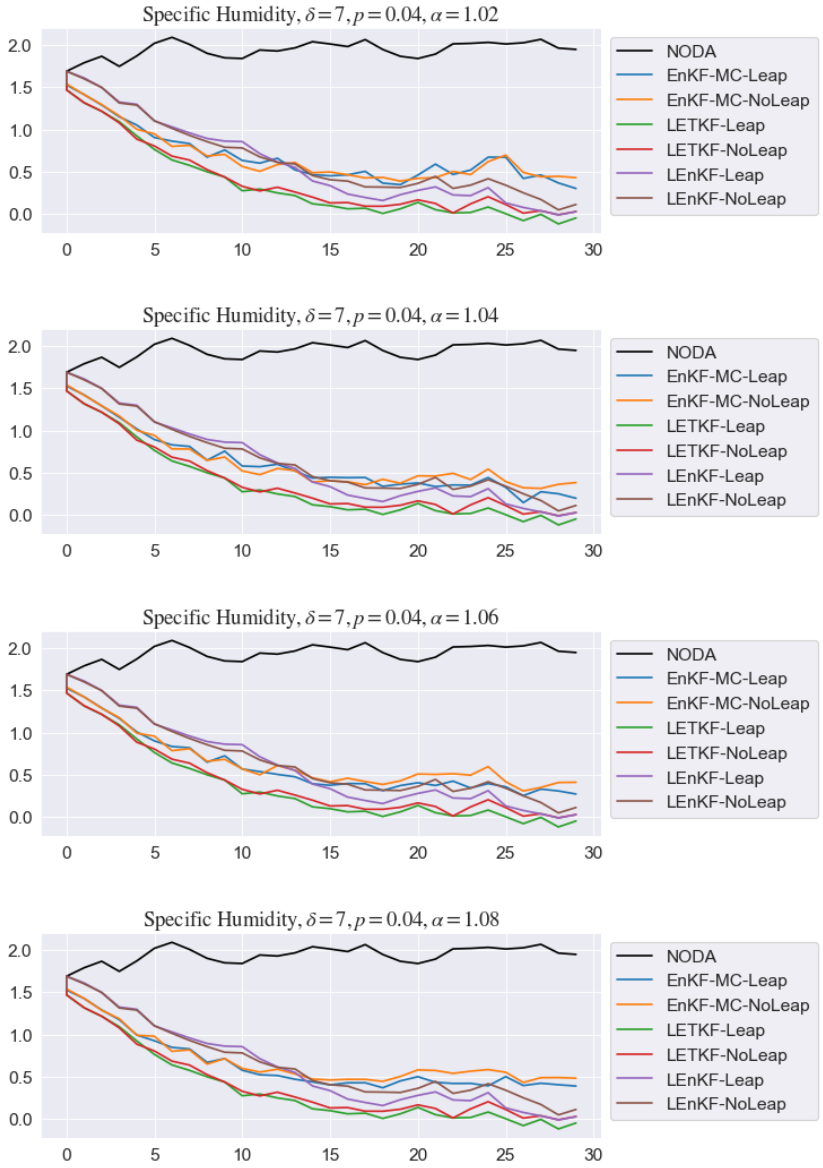


FIGURE 5.82: Time evolution of Specific Humidity for $\delta = 7$ and $p = 4\%$ varying α

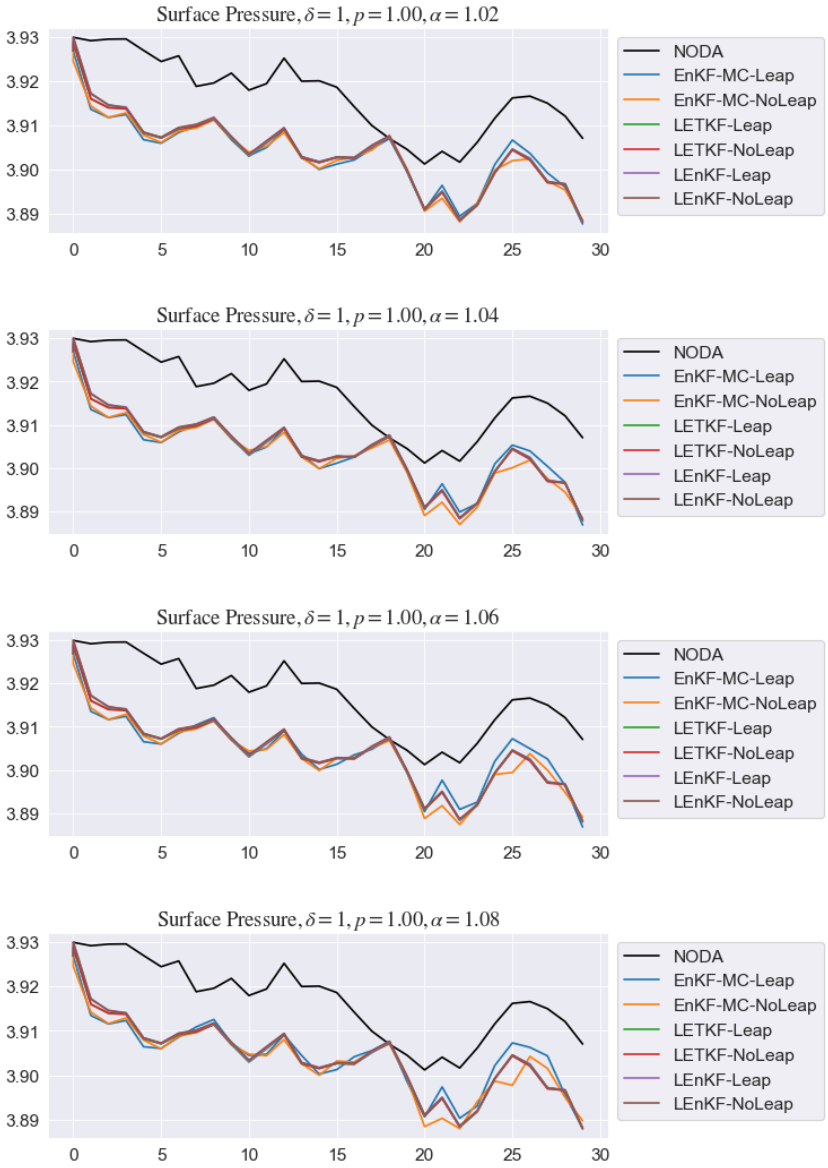


FIGURE 5.83: Time evolution of Surface Pressure for $\delta = 1$ and $p = 100\%$ varying α

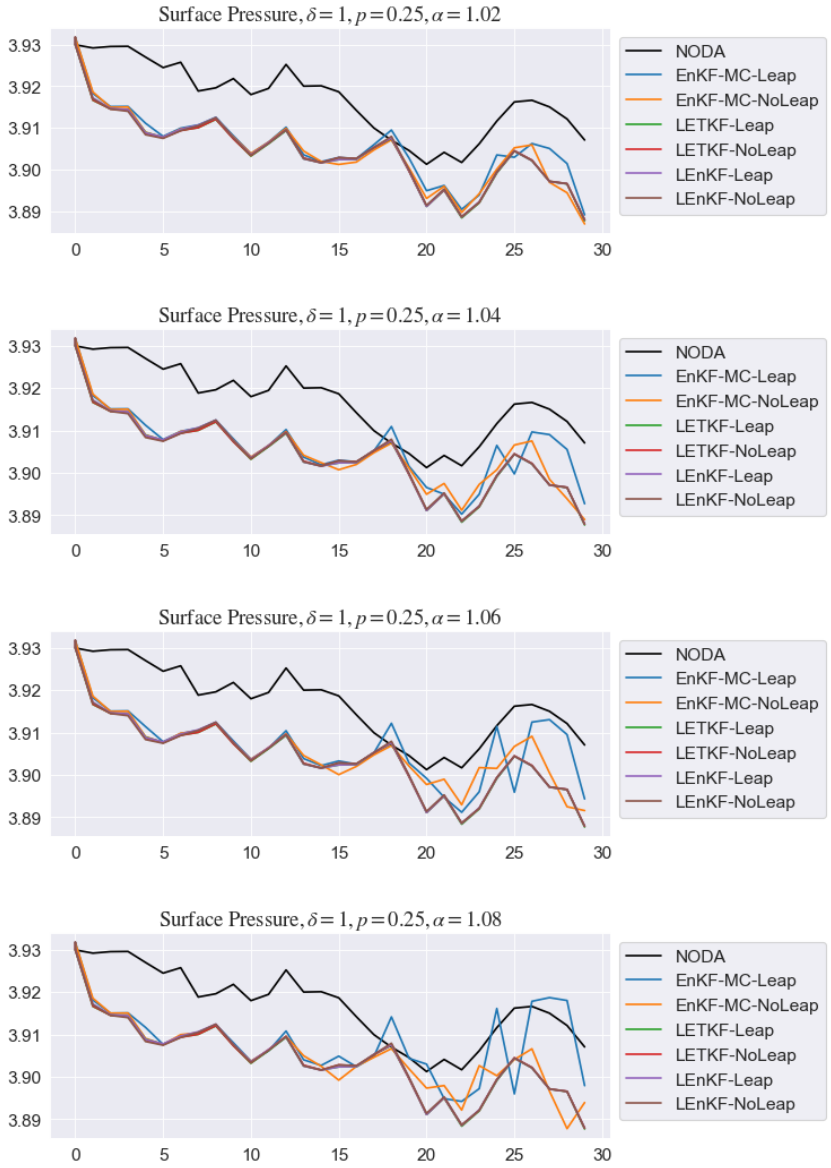


FIGURE 5.84: Time evolution of Surface Pressure for $\delta = 1$ and $p = 25\%$ varying α

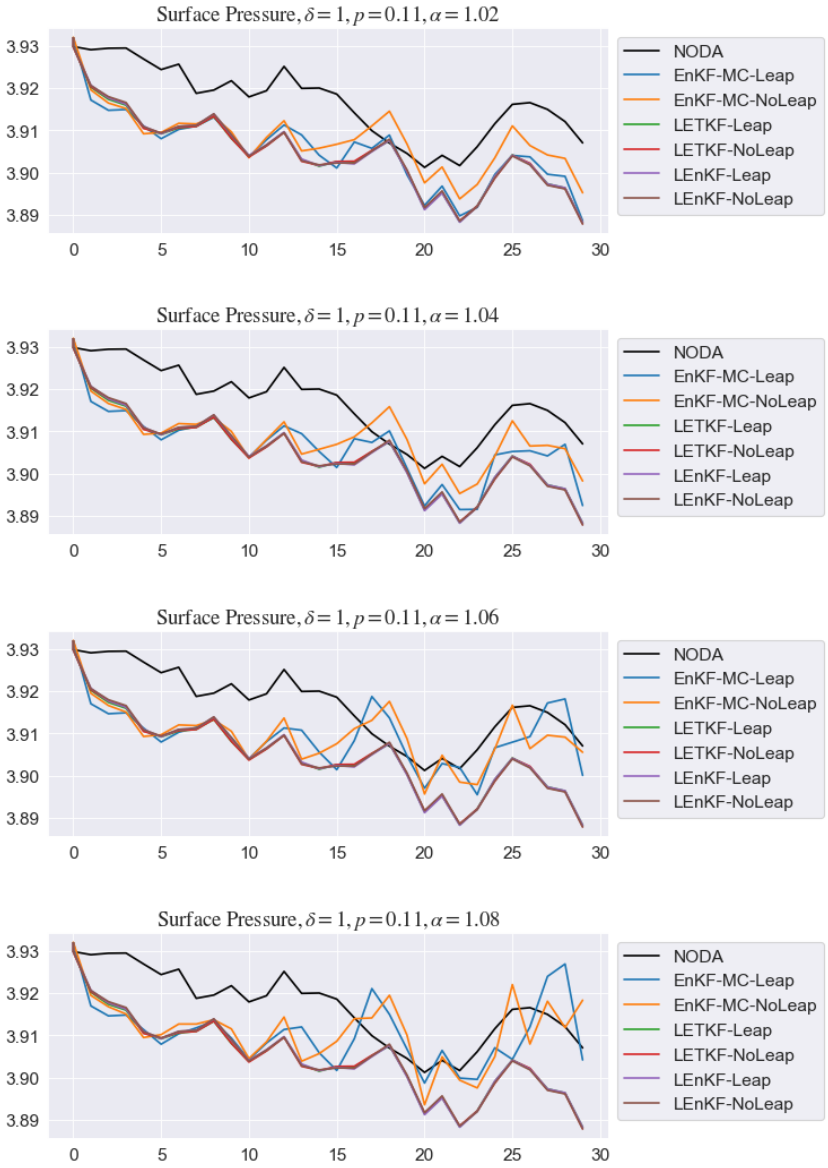


FIGURE 5.85: Time evolution of Surface Pressure for $\delta = 1$ and $p = 11\%$ varying α

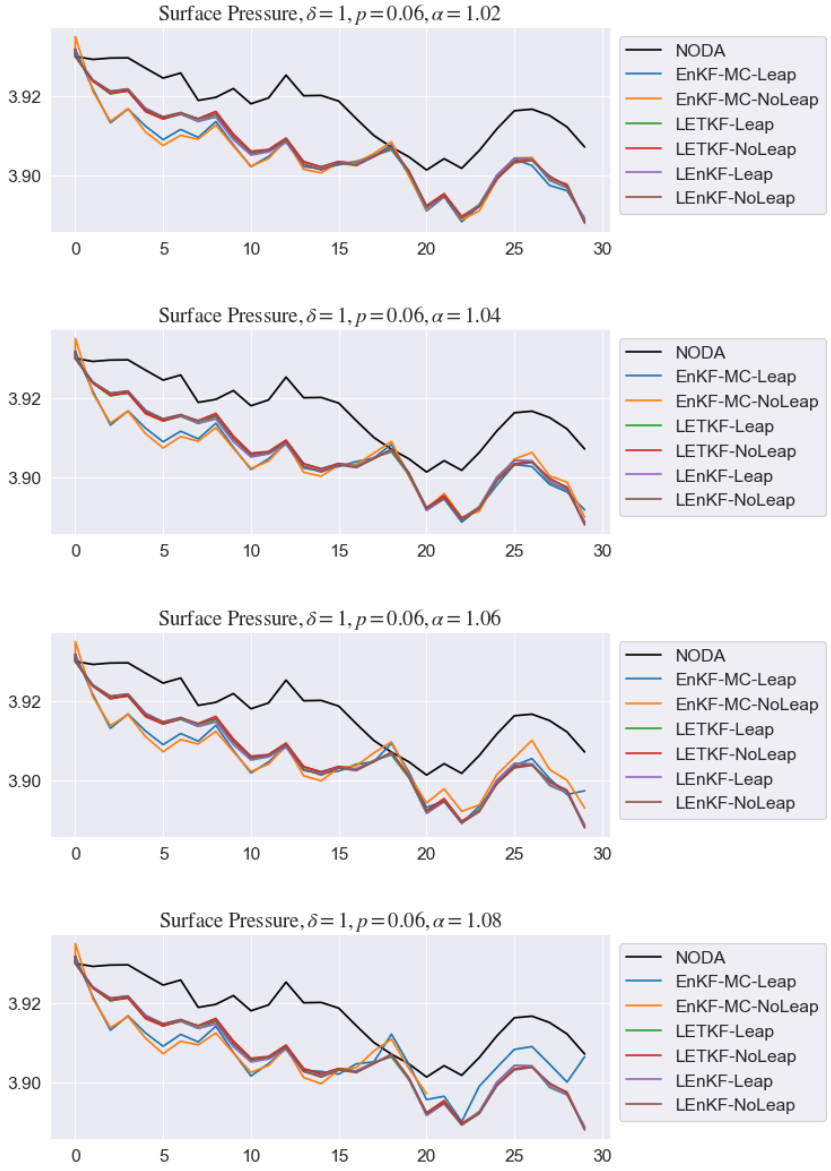


FIGURE 5.86: Time evolution of Surface Pressure for $\delta = 1$ and $p = 6\%$ varying α

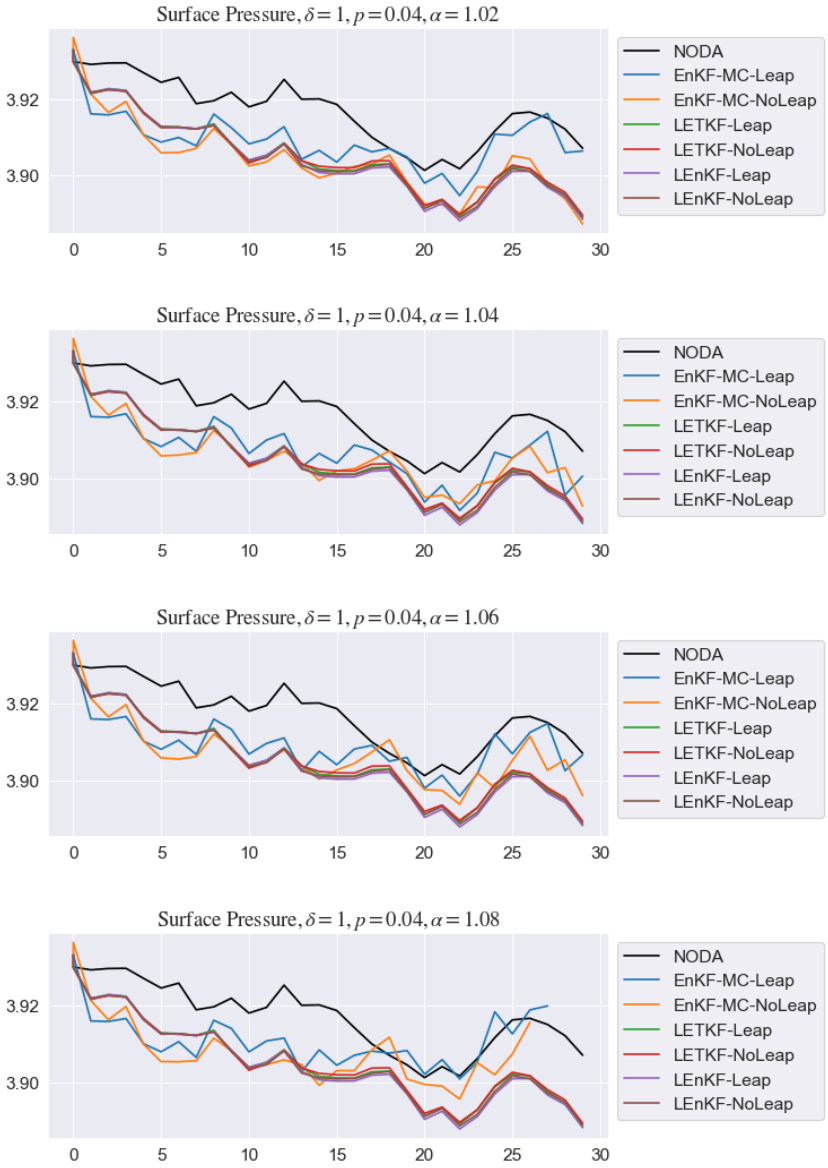


FIGURE 5.87: Time evolution of Surface Pressure for $\delta = 1$ and $p = 4\%$ varying α

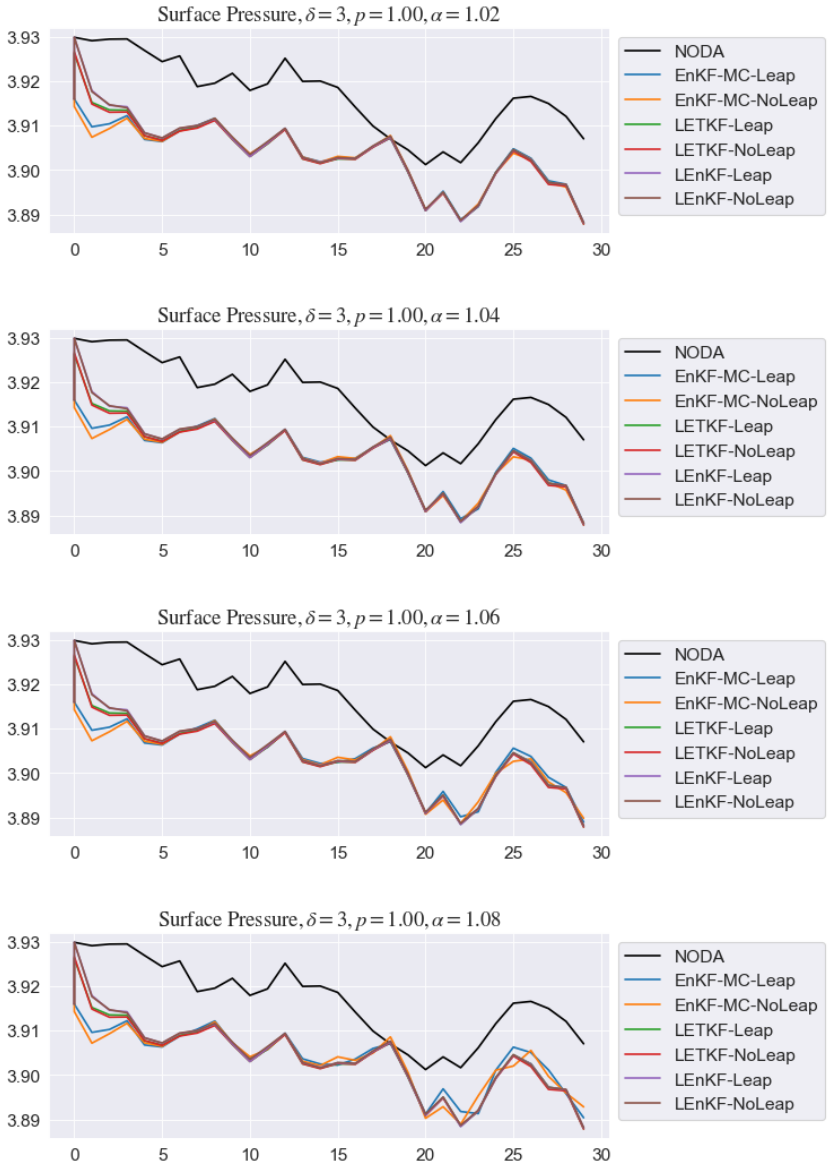


FIGURE 5.88: Time evolution of Surface Pressure for $\delta = 3$ and $p = 100\%$ varying α

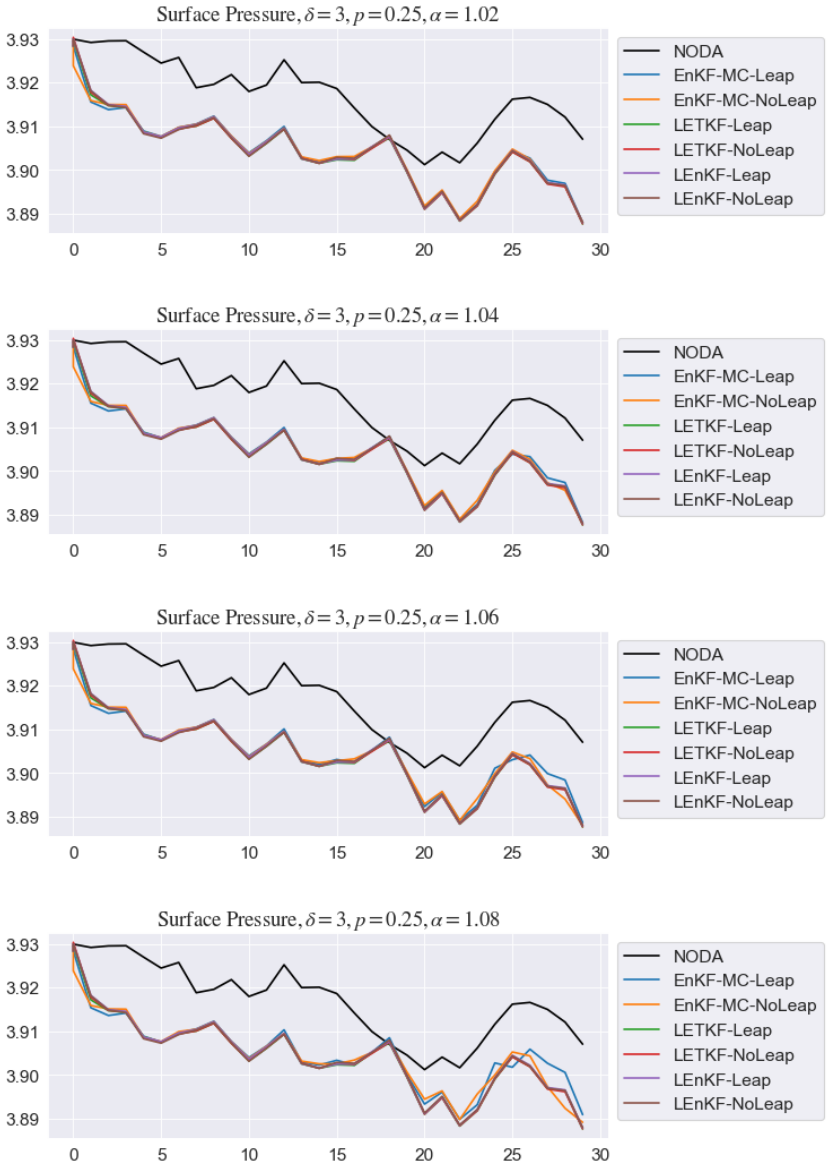


FIGURE 5.89: Time evolution of Surface Pressure for $\delta = 3$ and $p = 25\%$ varying α

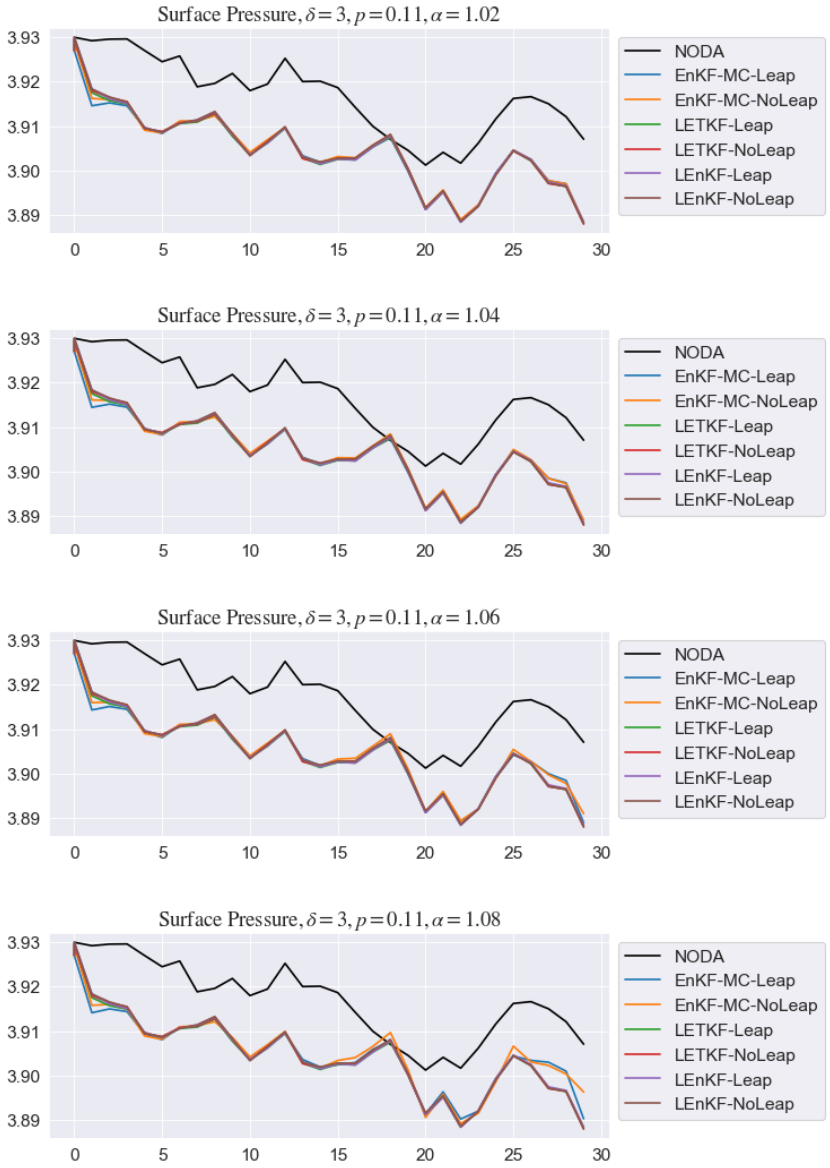


FIGURE 5.90: Time evolution of Surface Pressure for $\delta = 3$ and $p = 11\%$ varying α

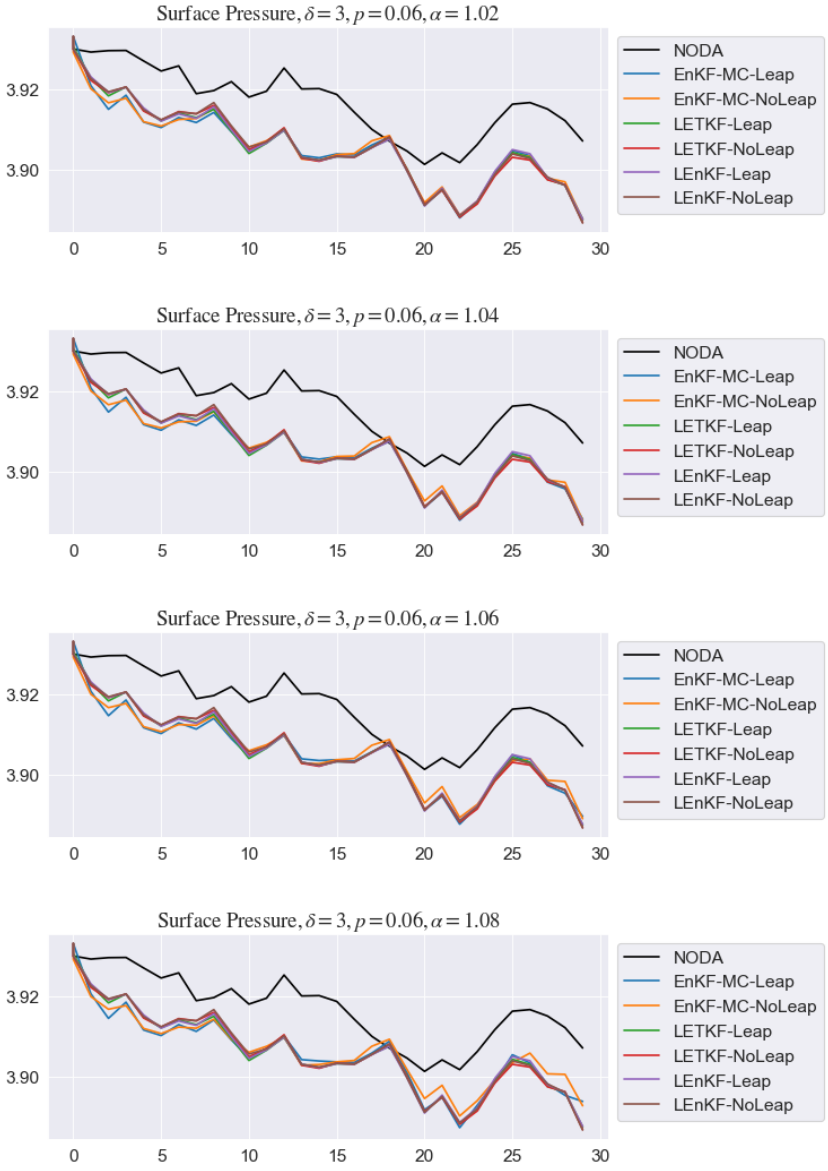


FIGURE 5.91: Time evolution of Surface Pressure for $\delta = 3$ and $p = 6\%$ varying α

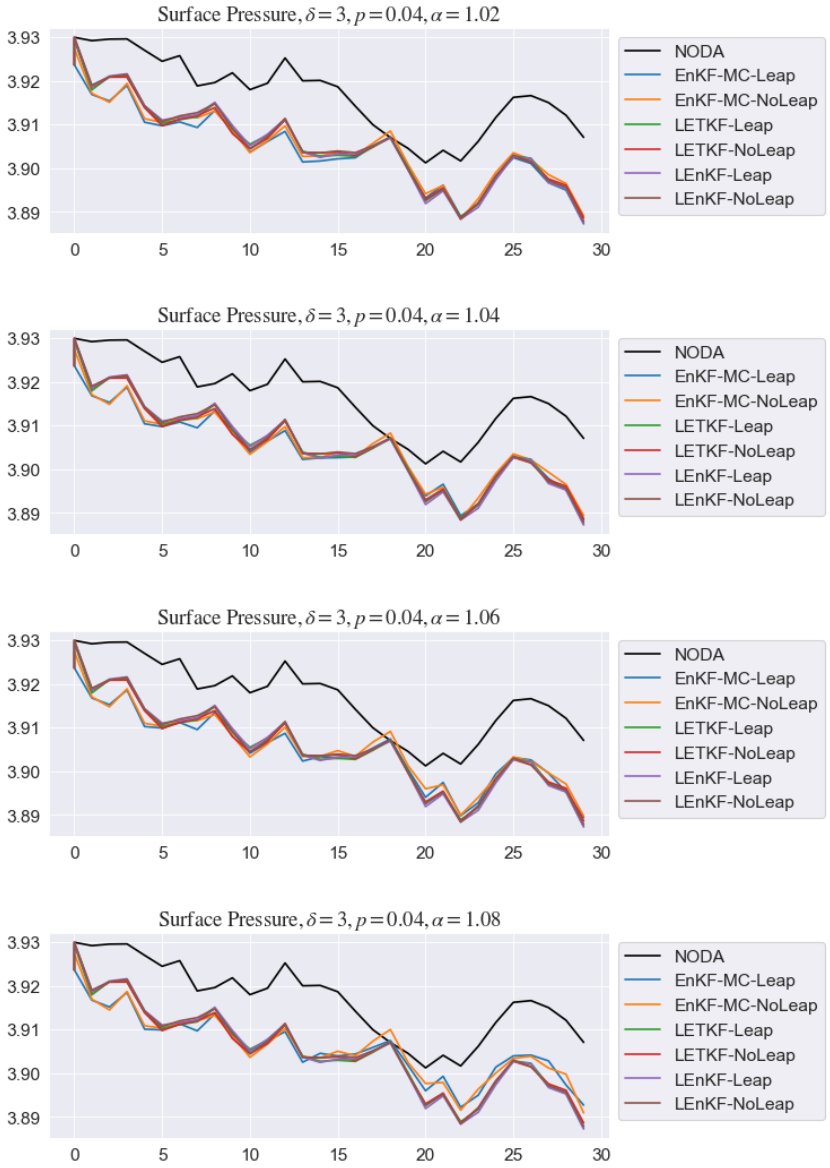


FIGURE 5.92: Time evolution of Surface Pressure for $\delta = 3$ and $p = 4\%$ varying α

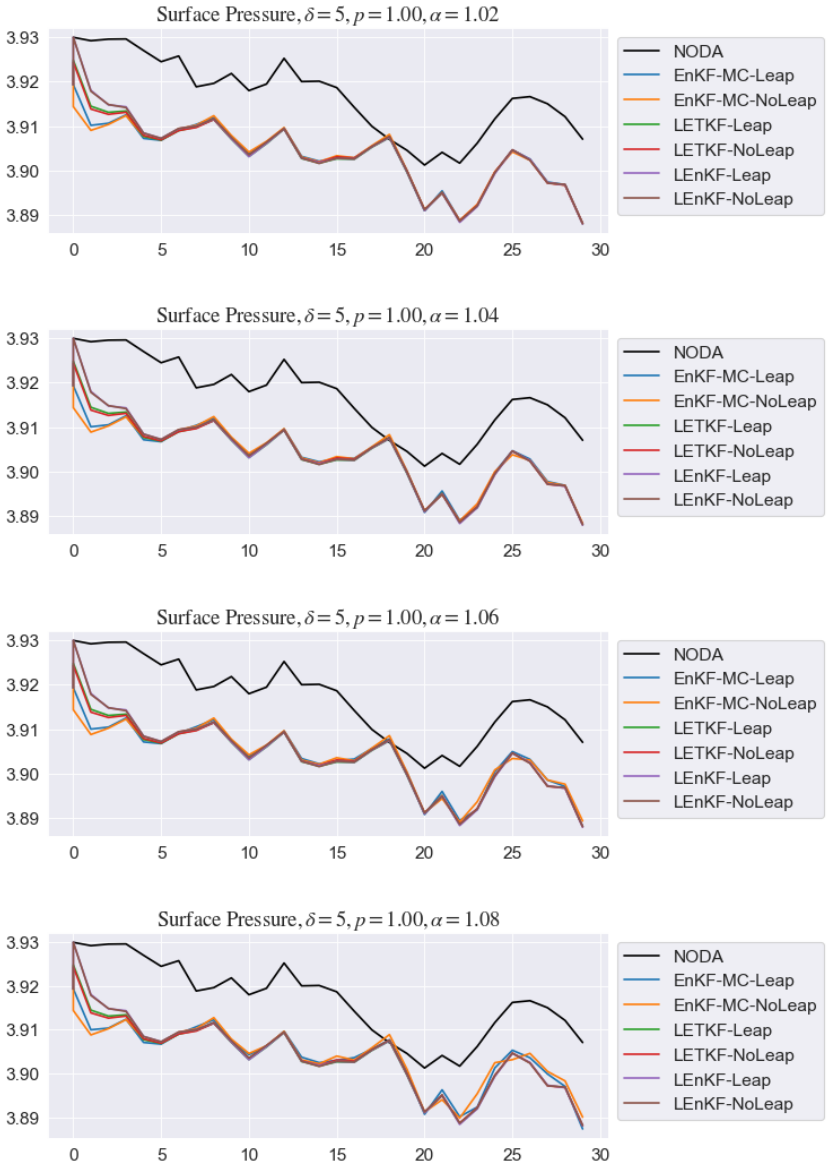


FIGURE 5.93: Time evolution of Surface Pressure for $\delta = 5$ and $p = 100\%$ varying α

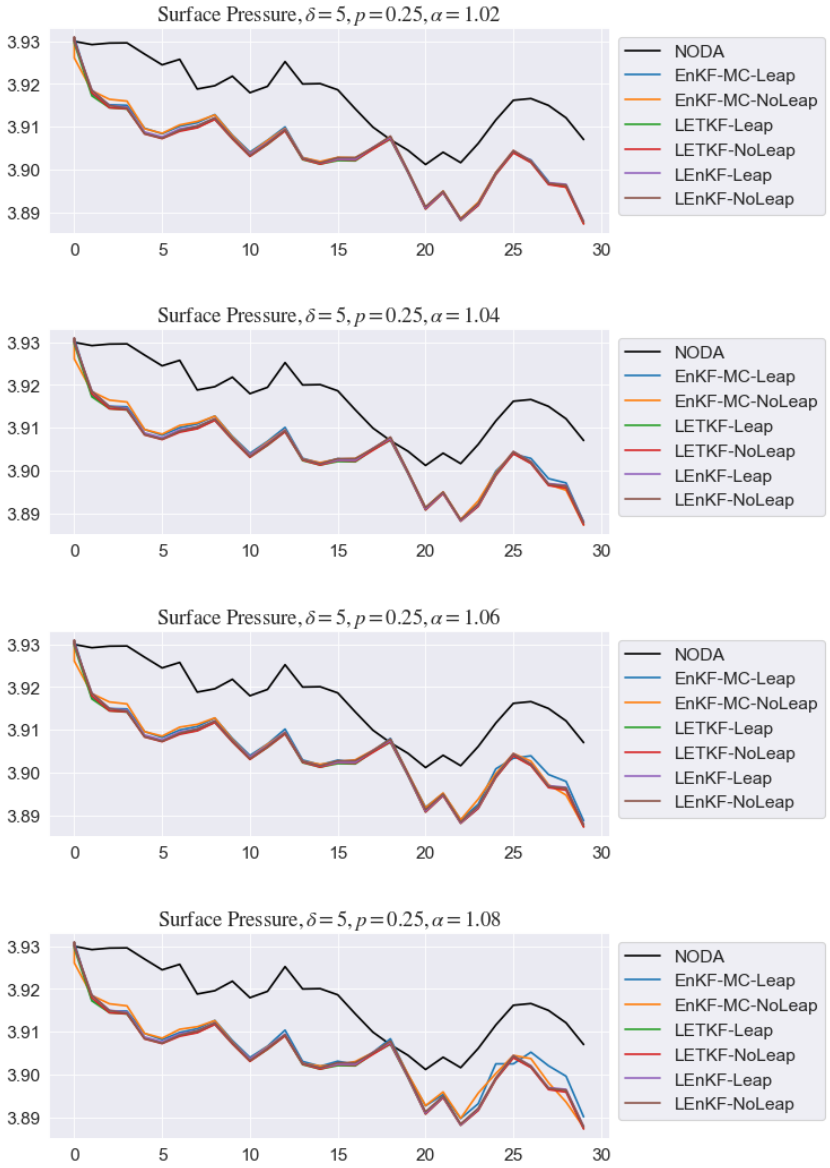


FIGURE 5.94: Time evolution of Surface Pressure for $\delta = 5$ and $p = 25\%$ varying α

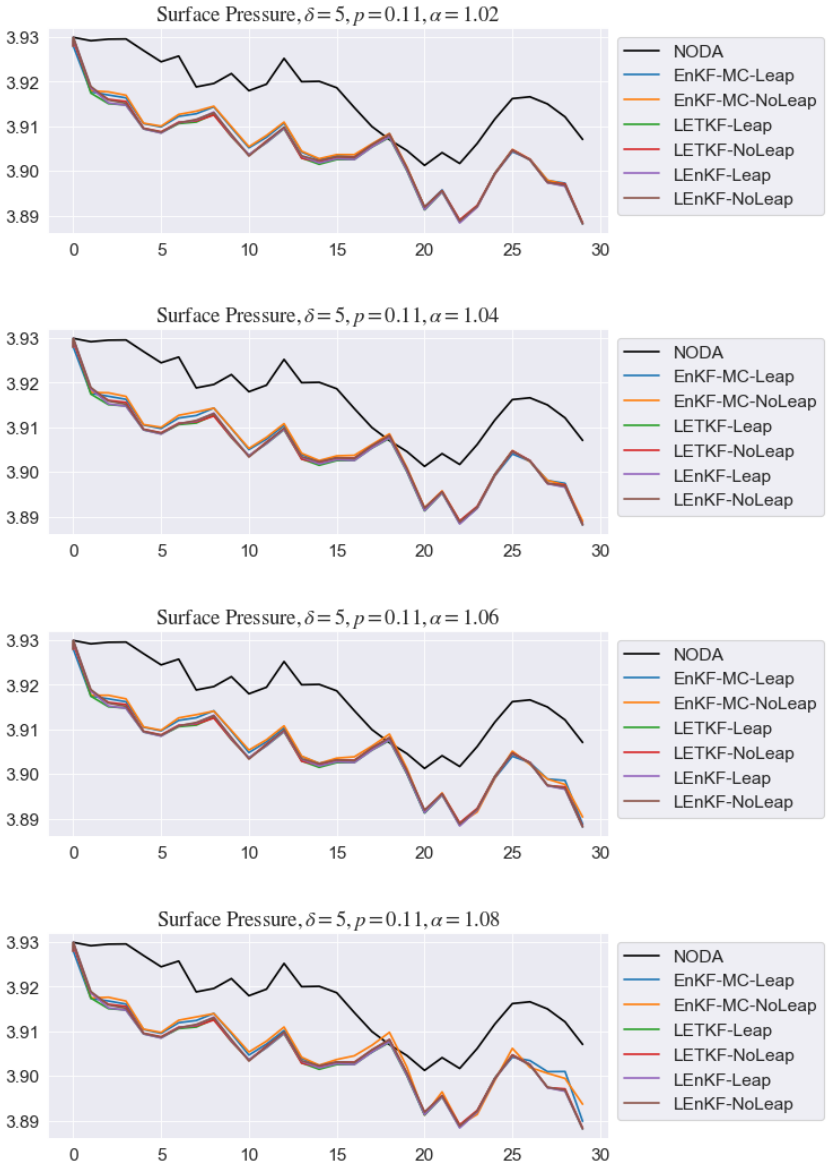


FIGURE 5.95: Time evolution of Surface Pressure for $\delta = 5$ and $p = 11\%$ varying α

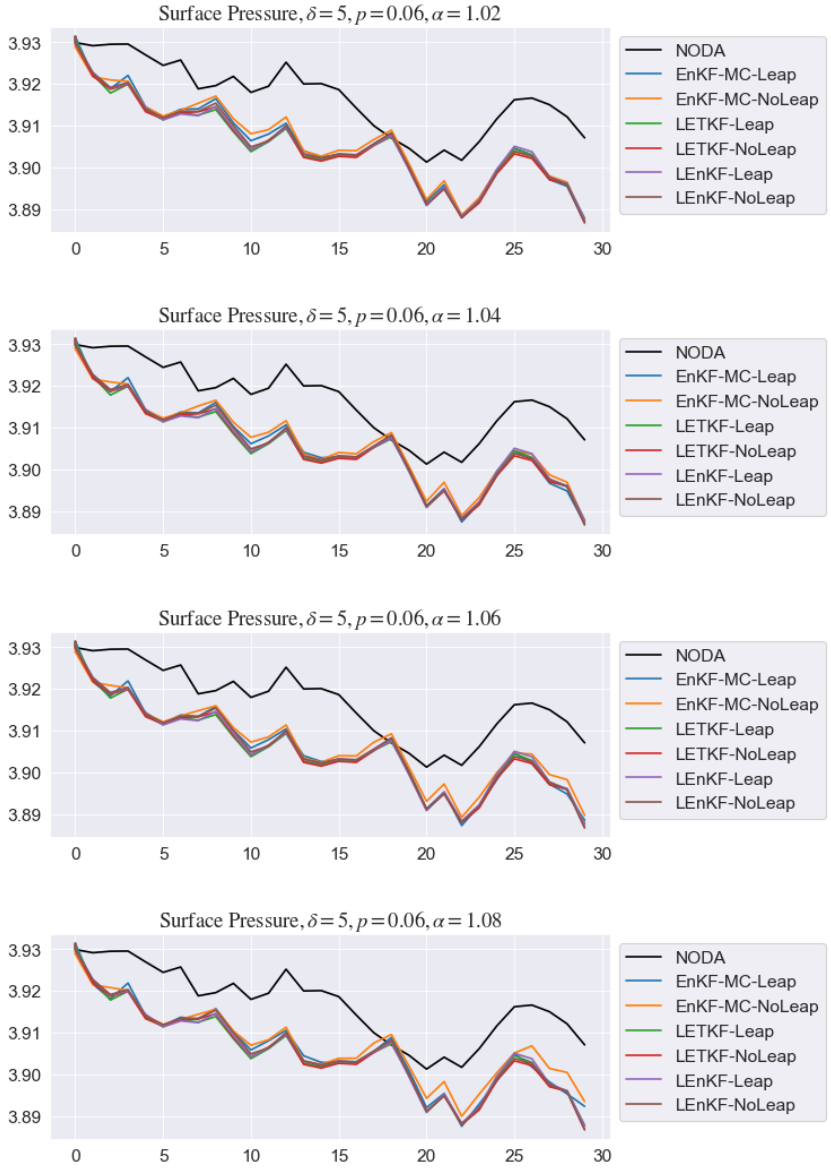


FIGURE 5.96: Time evolution of Surface Pressure for $\delta = 5$ and $p = 6\%$ varying α

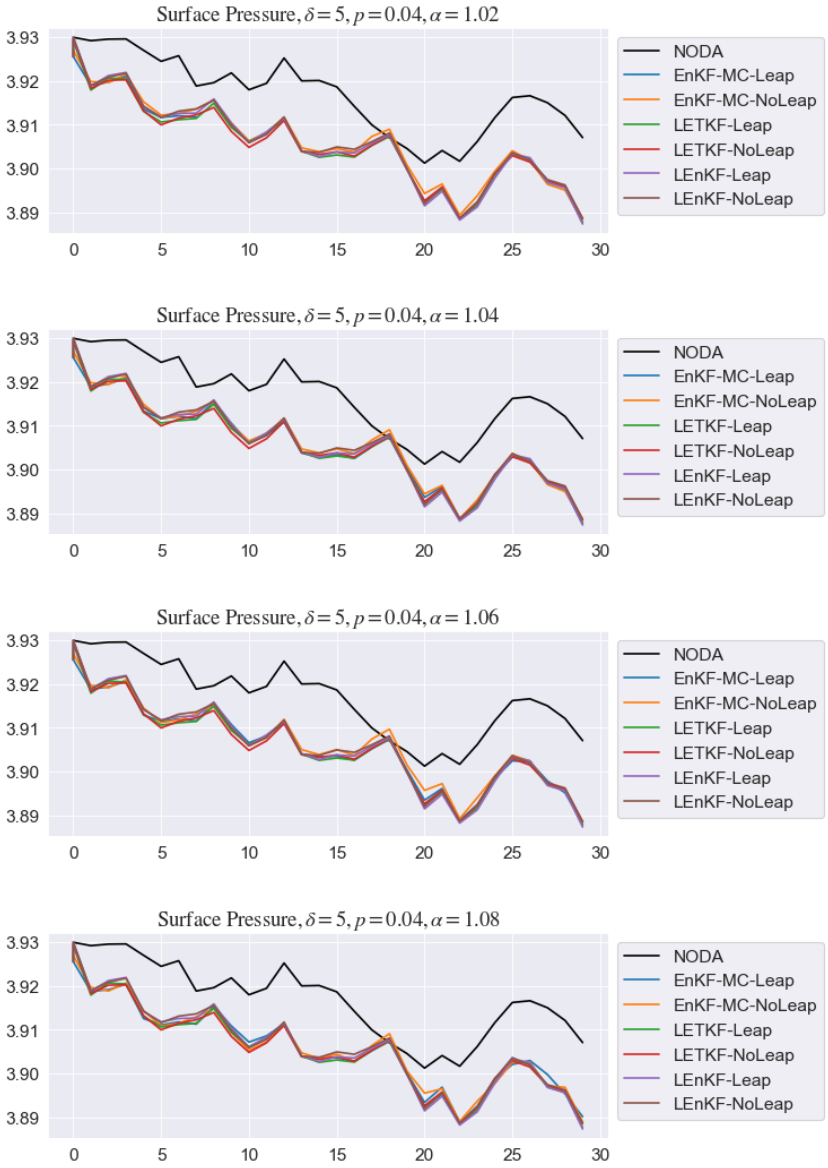


FIGURE 5.97: Time evolution of Surface Pressure for $\delta = 5$ and $p = 4\%$ varying α

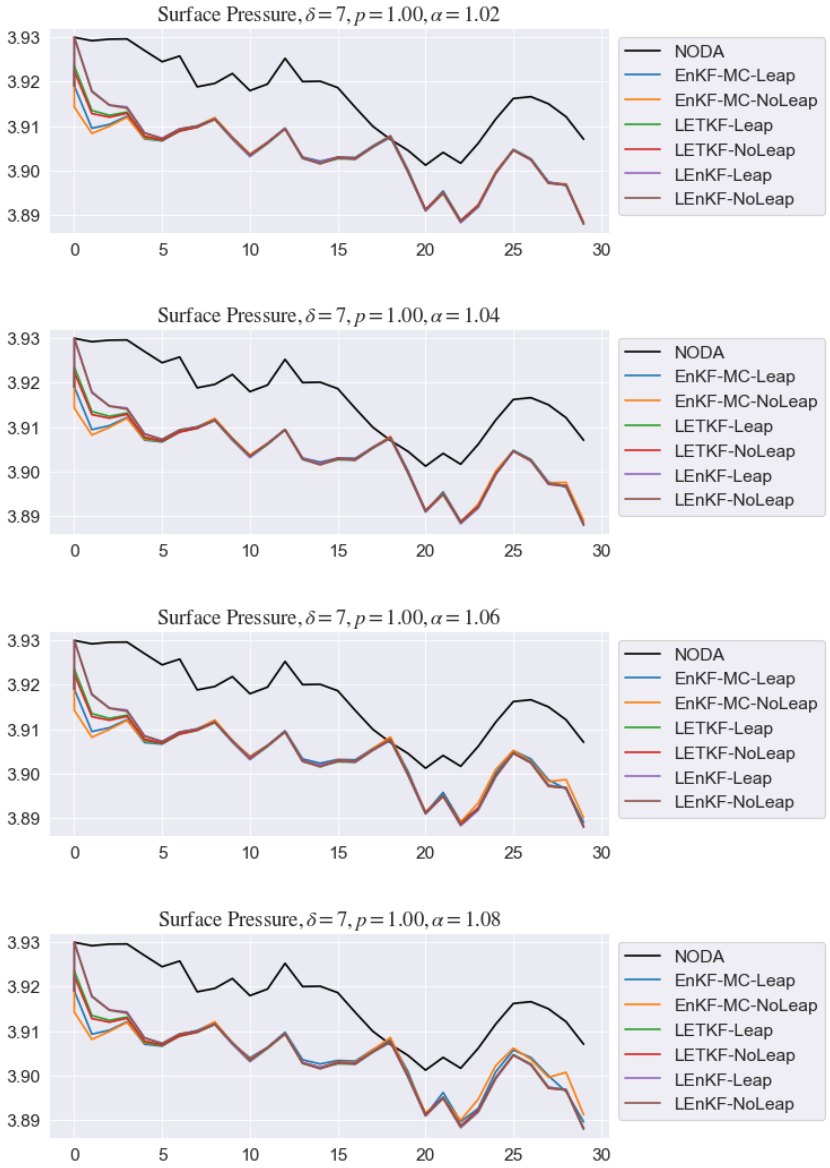


FIGURE 5.98: Time evolution of Surface Pressure for $\delta = 7$ and $p = 100\%$ varying α

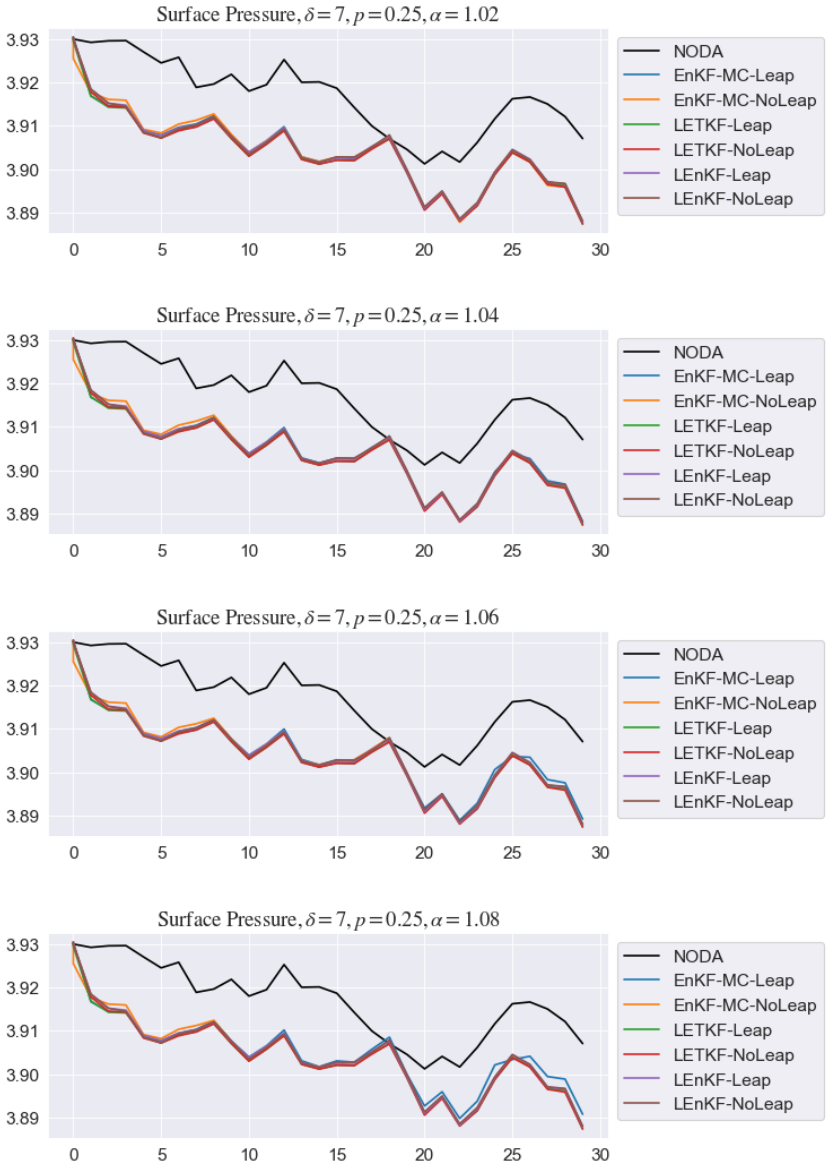


FIGURE 5.99: Time evolution of Surface Pressure for $\delta = 7$ and $p = 25\%$ varying α

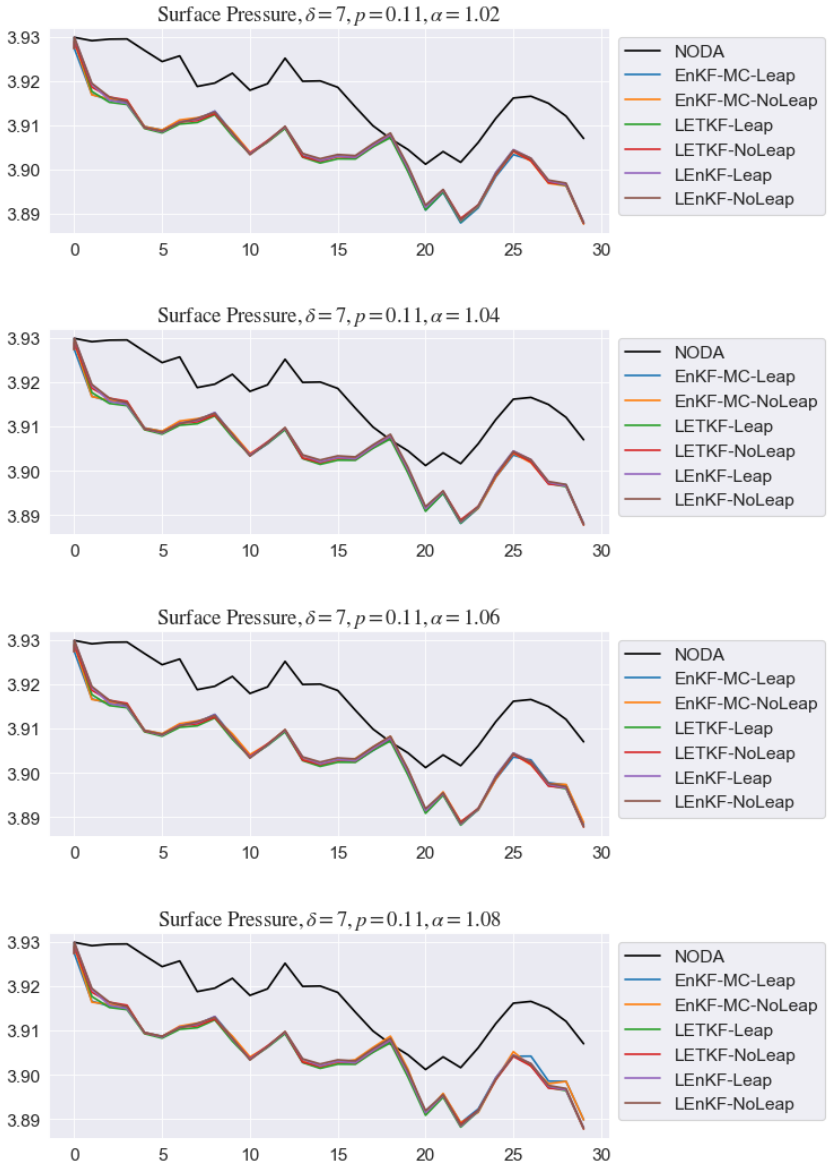


FIGURE 5.100: Time evolution of Surface Pressure for $\delta = 7$ and $p = 11\%$ varying α

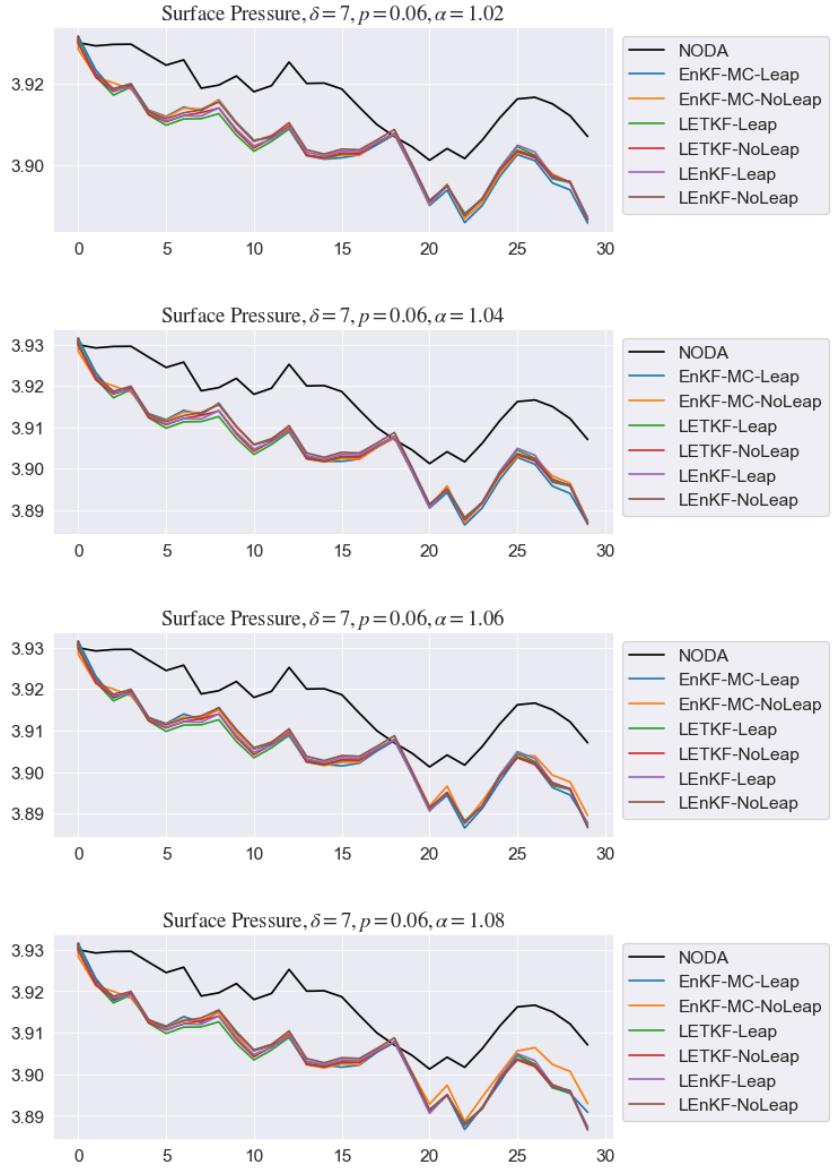


FIGURE 5.101: Time evolution of Surface Pressure for $\delta = 7$ and $p = 6\%$ varying α

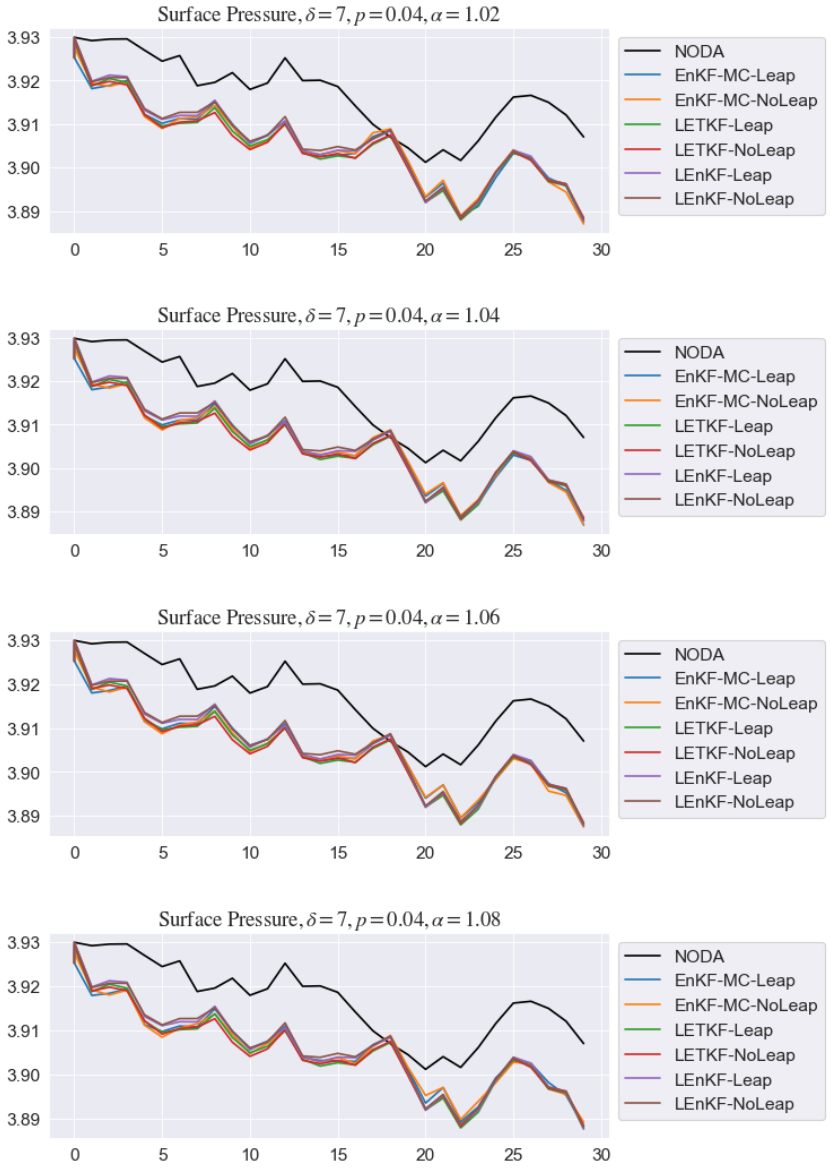
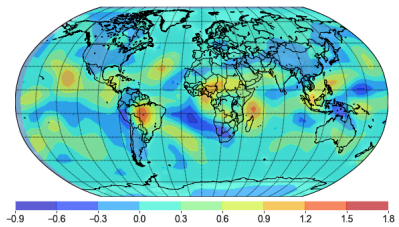
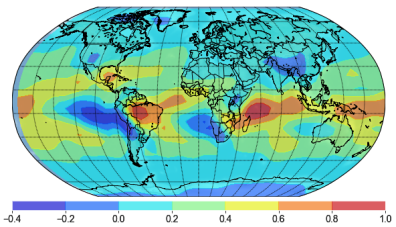


FIGURE 5.102: Time evolution of Surface Pressure for $\delta = 7$ and $p = 4\%$ varying α

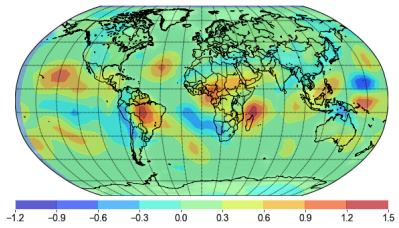
In Figure 5.103 snapshots of the specific humidity for different formulation at $\delta = 3$, $p = 11\%$ and $\alpha = 1.02$ are shown, at the third layer and first assimilation time.



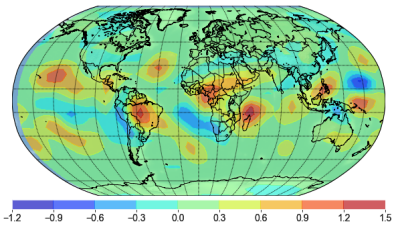
(A) Reference



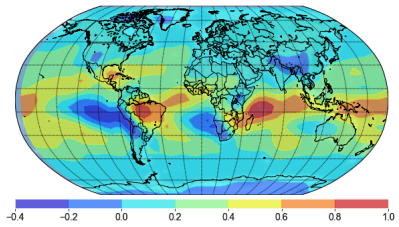
(B) Background



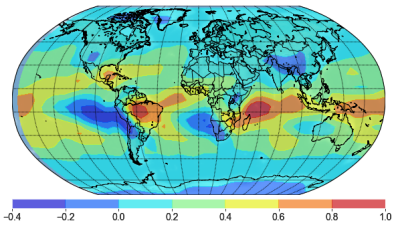
(C) EnKF-MC NoLeap



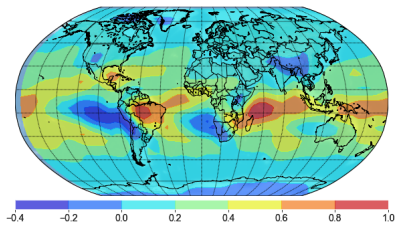
(D) EnKF-MC Leap



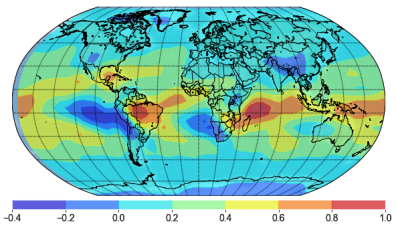
(E) LETKF NoLeap



(F) LETKF Leap



(G) LEnKF NoLeap



(H) LEnKF Leap

FIGURE 5.103: Snapshots for Specific Humidity with $\delta = 3$ and $p = 11\%$

In terms of RMSE, assimilation methods achieve accurate results for specific humidity, with no statistical difference in our proposal, using the sparse network we choose. Yet, surface pressure shows no difference regarding which method, and behave as well as the model itself, this can be by the normality assumption made in the formulation, as well as it has only one layer of data. Only the EnKF-MC method uses inflation in its formulation.

Chapter 6

Conclusions

This paper develops a comparison of three efficient formulation of the Ensemble Kalman Filter such as the LETKF, LEnKF and the EnKF-MC, using an Atmospheric General Circulation Model as SPEEDY, and showed how much the leaping step affects the assimilation process for each of them. We proposed an scheme which does not explicitly updates the leaping step and let the model itself to update it using the model dynamics, achieving similar results as using the leaping step in the update computation. This is of high relevance, as in operative scenarios, we can not afford to find a analytical method to solve the model's right hand side as, most of the time, no explicit formula is obtained. Even more, to obtain a real observation at such time can be expensive or even prohibitive, so methods to reduce the impact of such uncertainty are of high value.

We, as well, developed a test-suit to build sequential data assimilation methods and to test (and compare) multiple sequential data assimilation methods. By default, this package is released with five spectral resolutions of the SPEEDY model. The model implements the three sequential data assimilation formulations used in this proposal: the local ensemble Kalman filter, the local ensemble transform Kalman filter, and the ensemble Kalman filter based on a modified Cholesky decomposition onto the observation space. The package brings the flexibility to implement other methods as needed.

We compare the results using the RMSE for each of them and observed how similar their performance was, using localization schemes. The results reveal that, the accuracy in terms of root-mean-square-error of the proposed method is similar to that of one of using the leaping step, for different model configurations as observation percentage, localization radii and inflation.

There is also an opportunity to extend this package and to incorporate variational data assimilation methods. In future releases of our package, we expect to have ensemble-based variational data assimilation methods to have another manner to digest observations in assimilation stages.

Acknowledgments

This work was supported by the Applied Math and Computer Science Laboratory (AML-CS) at Universidad del Norte.

Bibliography

Abaza, Mabrouk et al. (2017). "On the incidence of meteorological and hydrological processors: effect of resolution, sharpness and reliability of hydrological ensemble forecasts". In: *Journal of Hydrology*.

Amezcua, Javier, Eugenia Kalnay, and Paul D Williams (2011). "The effects of the RAW filter on the climatology and forecast skill of the SPEEDY model". In: *Monthly Weather Review* 139.2, pp. 608–619.

Anderson, Jeffrey L (2001). "An ensemble adjustment Kalman filter for data assimilation". In: *Monthly weather review* 129.12, pp. 2884–2903.

— (2019). "A Nonlinear Rank Regression Method for Ensemble Kalman Filter Data Assimilation". In: *Monthly Weather Review* 147.8, pp. 2847–2860.

Anderson, Jeffrey L. and S. L. Anderson (1999). "A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts". In: *Monthly Weather Review* 127, pp. 2741–2758.

Barker, D. M. et al. (2004). "A Three-Dimensional Variational Data Assimilation System for MM5: Implementation and Initial Results". In: *Monthly Weather Review* 132.4, pp. 897 –914. DOI: 10.1175/1520-0493(2004)132<0897:ATVDAS>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/mwre/132/4/1520-0493_2004_132_0897_atvdas_2.0.co_2.xml.

Bauer, Peter, Alan Thorpe, and Gilbert Brunet (Sept. 2015). "The quiet revolution of numerical weather prediction". In: *Nature* 525, pp. 47–55. DOI: 10.1038/nature14956.

Berger, James O. and Leonard A. Smith (2019). "On the Statistical Formalism of Uncertainty Quantification". In: *Annual Review of Statistics and Its Application* 6.1, pp. 433–460. DOI: 10.1146/annurev-statistics-030718-105232. eprint: https://doi.org/10.1146/annurev-statistics-030718-105232. URL: https://doi.org/10.1146/annurev-statistics-030718-105232.

Bickel, Peter J, Elizaveta Levina, et al. (2008). "Regularized estimation of large covariance matrices". In: *The Annals of Statistics* 36.1, pp. 199–227.

Bracco, Annalisa et al. (2004). "Internal variability, external forcing and climate trends in multi-decadal AGCM ensembles". In: *Climate Dynamics* 23.6, pp. 659–678.

Buehner, Mark (2011). "Evaluation of a Spatial/Spectral Covariance Localization Approach for Atmospheric Data Assimilation". In: *Monthly Weather Review* 140.2, pp. 617–636. DOI: 10.1175/MWR-D-10-05052.1.

Chen, Zhe (2003). "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond". In: *Statistics* 182. DOI: 10.1080/02331880309257.

Dovera, Laura and Ernesto Della Rossa (2011). "Multimodal ensemble Kalman filtering using Gaussian mixture models". In: *Computational Geosciences* 15.2, pp. 307–323.

Evensen, Geir (1994). "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In: *Journal of Geophysical Research: Oceans* 99.C5, pp. 10143–10162.

— (2006). *Data Assimilation: The Ensemble Kalman Filter*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 354038300X.

Evensen, Geir (2009). "The ensemble Kalman filter for combined state and parameter estimation". In: *IEEE Control Systems* 29.3.

Fan, Jianqing, Yuan Liao, and Han Liu (2016). "An overview of the estimation of large covariance and precision matrices". In: *The Econometrics Journal* 19.1, pp. C1–C32. DOI: https://doi.org/10.1111/ectj.12061. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ectj.12061. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/ectj.12061.

Gillijns, S et al. (2006). "What is the Ensemble Kalman Filter and How Well Does It Work?" In: *American Control Conference, 2006*, 6 pp.–. DOI: 10.1109/ACC.2006.1657419.

Godinez, Humberto C and J David Moulton (2012). "An efficient matrix-free algorithm for the ensemble Kalman filter". In: *Computational Geosciences* 16.3, pp. 565–575.

Gustafsson, Nils and J Bojarova (2014). "Four-dimensional ensemble variational (4D-En-Var) data assimilation for the high resolution limited area model (HIRLAM)". In: *Nonlinear Processes in Geophysics* 21.4, pp. 745–762.

Hamill, Thomas M, Jeffrey S Whitaker, and Chris Snyder (2001). "Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter". In: *Monthly Weather Review* 129.11, pp. 2776–2790.

Han, Yunfeng, Jucheng Zhang, and Dajun Sun (2018). "Error control and adjustment method for underwater wireless sensor network localization". In: *Applied Acoustics* 130, pp. 293–299.

Houtekamer, Peter L and Herschel L Mitchell (1998). "Data assimilation using an ensemble Kalman filter technique". In: *Monthly Weather Review* 126.3, pp. 796–811.

Hunt, Brian R., Eric J. Kostelich, and Istvan Szunyogh (2007). "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter". In: *Physica D: Nonlinear Phenomena* 230.1. Data Assimilation, pp. 112–126. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2006.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0167278906004647.

Hurrell, J.W., C. Deser, and A.S. Phillips (2019). "North Atlantic Oscillation (NAO)". In: *Encyclopedia of Ocean Sciences (Third Edition)*. Ed. by J. Kirk Cochran, Henry J. Bokuniewicz, and Patricia L. Yager. Third Edition. Oxford: Academic Press, pp. 447–454. ISBN: 978-0-12-813082-7. DOI: https://doi.org/10.1016/B978-0-12-409548-9.11621-5. URL: https://www.sciencedirect.com/science/article/pii/B9780124095489116215.

Ito, Shin-ichi et al. (2016). "Data assimilation for massive autonomous systems based on a second-order adjoint method". In: *Physical Review E* 94.4, p. 43307.

Jones, Philip W., Kevin Hamilton, and R. John Wilson (1997). "A Very High Resolution General Circulation Model Simulation of the Global Circulation in Austral Winter". In: *Journal of the Atmospheric Sciences* 54.8, pp. 1107 –1116. DOI: 10.1175/1520-0469(1997)054<1107:AVHRGC>2.0.CO;2.

Kalman, R. E. (Mar. 1960). "A new approach to linear filtering and prediction problems". In: *Journal of Fluids Engineering, Transactions of the ASME* 82.1, pp. 35–45. ISSN: 1528901X. DOI: 10.1115/1.3662552. URL: https://doi.org/10.1115/1.3662552.

Kalman, R. E. and R. S. Bucy (Mar. 1961). "New results in linear filtering and prediction theory". In: *Journal of Fluids Engineering, Transactions of the ASME* 83.1, pp. 95–108. ISSN: 1528901X. DOI: 10.1115/1.3658902. URL: https://doi.org/10.1115/1.3658902.

Kalnay, Eugenia (2002). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press. DOI: 10.1017/CBO9780511802270.

Kondo, Keiichi and Takemasa Miyoshi (2016). "Impact of Removing Covariance Localization in an Ensemble Kalman Filter: Experiments with 10 240 Members Using an Intermediate AGCM". In: *Monthly Weather Review* 144.12, pp. 4849–4865. DOI: 10.1175/MWR-D-15-0388.1. URL: https://doi.org/10.1175/MWR-D-15-0388.1.

Kotsuki, Shunji, Yousuke Sato, and Takemasa Miyoshi (2020). "Data Assimilation for Climate Research: Model Parameter Estimation of Large-Scale Condensation Scheme". In: *Journal of Geophysical Research: Atmospheres* 125.1. e2019JD031304 2019JD031304, e2019JD031304. DOI: https://doi.org/10.1029/2019JD031304. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019JD031304. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JD031304.

Kucharski, Fred, Franco Molteni, and Annalisa Bracco (2006). "Decadal interactions between the western tropical Pacific and the North Atlantic Oscillation". In: *Climate dynamics* 26.1, pp. 79–91.

Kwiatkowski, Evan and Jan Mandel (Apr. 2015). "Convergence of the Square Root Ensemble Kalman Filter in the Large Ensemble Limit". In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1, pp. 1–17. DOI: 10.1137/140965363. URL: http://arxiv.org/abs/1404.4093http://dx.doi.org/10.1137/140965363.

Lee, Yoonsang, Andrew J Majda, and Di Qi (2016). "Preventing catastrophic filter divergence using adaptive additive inflation for baroclinic turbulence". In: *Monthly Weather Review* 2016.

Lei, Lili and Jeffrey S Whitaker (2017). "Evaluating the tradeoffs between ensemble size and ensemble resolution in an ensemble-variational data assimilation system". In: *Journal of Advances in Modeling Earth Systems*.

Lei, Lili, Jeffrey S Whitaker, and Craig Bishop (2018). "Improving assimilation of radiance observations by implementing model space localization in an ensemble Kalman filter". In: *Journal of Advances in Modeling Earth Systems* 10.12, pp. 3221–3232.

Levina, Elizaveta, Adam Rothman, Ji Zhu, et al. (2008). "Sparse estimation of large covariance matrices via a nested lasso penalty". In: *The Annals of Applied Statistics* 2.1, pp. 245–263.

Liu, Zhenkun et al. (Nov. 2019). "A combined forecasting model for time series: Application to short-term wind speed forecasting". In: *Applied Energy*, p. 114137. ISSN: 0306-2619. DOI: 10.1016/J.APENERGY.2019.114137. URL: https://www.sciencedirect.com/science/article/abs/pii/S0306261919318240.

Lorenc, A. C. (1986). "Analysis methods for numerical weather prediction". In: *Quarterly Journal of the Royal Meteorological Society* 112.474, pp. 1177–1194. DOI: https://doi.org/10.1002/qj.49711247414. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.49711247414. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.49711247414.

Lorenc, Andrew C (2003a). "Modelling of error covariances by 4D-Var data assimilation". In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 129.595, pp. 3167–3182.

— (2003b). "The potential of the ensemble Kalman filter for NWP—a comparison with 4D-Var". In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 129.595, pp. 3183–3203.

Mandel, Jan (Jan. 2006). "Efficient Implementation of the Ensemble Kalman Filter". In:

Miyoshi, Takemasa (2011). "The Gaussian approach to adaptive covariance inflation and its implementation with the local ensemble transform Kalman filter". In: *Monthly Weather Review* 139.5, pp. 1519–1535.

Miyoshi, Takemasa, Keiichi Kondo, and Toshiyuki Imamura (2014). "The 10,240-member ensemble Kalman filtering with an intermediate AGCM". In: *Geophysical Research Letters* 41.14, pp. 5264–5271.

Molteni, Franco (Jan. 2003a). "Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments". In: *Climate Dynamics* 20, pp. 175–191. DOI: 10.1007/s00382-002-0268-2.

— (2003b). "Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments". In: *Climate dynamics* 20.2, pp. 175–191.

Nino-Ruiz, Elias, Adrian Sandu, and Xinwei Deng (May 2016). "An Ensemble Kalman Filter Implementation Based on Modified Cholesky Decomposition for Inverse Covariance Matrix Estimation". In: *SIAM Journal on Scientific Computing* 40. DOI: 10.1137/16M1097031.

Nino-Ruiz, Elias D (2018). "Implicit surrogate models for trust region based methods". In: *Journal of Computational Science*.

Nino-Ruiz, Elias D, Haiyan Cheng, and Rolando Beltran (2018). "A Robust Non-Gaussian Data Assimilation Method for Highly Non-Linear Models". In: *Atmosphere* 9.4, p. 126.

Nino-Ruiz, Elias D, Adrian Sandu, and Xinwei Deng (2017). "A parallel implementation of the ensemble Kalman filter based on modified Cholesky decomposition". In: *Journal of Computational Science*.

— (2018). "An ensemble Kalman filter implementation based on modified Cholesky decomposition for inverse covariance matrix estimation". In: *SIAM Journal on Scientific Computing* 40.2, A867–A886.

Nino Ruiz, EliasD., Adrian Sandu, and Jeffrey Anderson (2014). "An Efficient Implementation of the Ensemble Kalman Filter Based on an Iterative Sherman–Morrison Formula". In: *Statistics and Computing*, pp. 1–17. ISSN: 0960-3174. DOI: 10.1007/s11222-014-9454-4. URL: http://dx.doi.org/10.1007/s11222-014-9454-4.

Ott, Edward et al. (2004a). "A local ensemble Kalman filter for atmospheric data assimilation". In: *Tellus A: Dynamic Meteorology and Oceanography* 56.5, pp. 415–428. DOI: 10.3402/tellusa.v56i5.14462. eprint: https://doi.org/10.3402/tellusa.v56i5.14462. URL: https://doi.org/10.3402/tellusa.v56i5.14462.

— (2004b). "A local ensemble Kalman filter for atmospheric data assimilation". In: *Tellus A: Dynamic Meteorology and Oceanography* 56.5, pp. 415–428. DOI: 10.3402/tellusa.v56i5.14462. eprint: https://doi.org/10.3402/tellusa.v56i5.14462. URL: https://doi.org/10.3402/tellusa.v56i5.14462.

Pourahmadi, Mohsen (2011). "Covariance Estimation: The GLM and Regularization Perspectives". In: *Statistical Science* 26.3. ISSN: 0883-4237. DOI: 10.1214/11-sts358. URL: http://dx.doi.org/10.1214/11-STS358.

Putnam, Bryan J et al. (2017). "Ensemble Probabilistic Prediction of a Mesoscale Convective System and Associated Polarimetric Radar Variables using Single-Moment and Double-Moment Microphysics Schemes and EnKF Radar Data Assimilation". In: *Monthly Weather Review* 2017.

Rew, Russ and Glenn Davis (1990). "NetCDF: an interface for scientific data access". In: *IEEE computer graphics and applications* 10.4, pp. 76–82.

Roth, Michael et al. (Aug. 2017). "The Ensemble Kalman Filter: A Signal Processing Perspective". In: *EURASIP Journal on Advances in Signal Processing* 2017, p. 56. DOI: 10.1186/s13634-017-0492-x.

Ruiz, Juan, Manuel Pulido, and Takemasa Miyoshi (Apr. 2013). "Estimating Model Parameters with Ensemble-Based Data Assimilation: A Review". In: *Journal of the Meteorological Society of Japan* 91. DOI: 10.2151/jmsj.2013-201.

Saetrom, Jon and Henning Omre (Dec. 2013). "Uncertainty Quantification in the Ensemble Kalman Filter". In: *Scandinavian Journal of Statistics* 40. DOI: 10.1111/sjos.12039.

Sakov, Pavel, Geir Evensen, and Laurent Bertino (2010). "Asynchronous data assimilation with the EnKF". In: *Tellus A* 62.1, pp. 24–29.

Stengel, M et al. (2009). "Assimilation of SEVIRI infrared radiances with HIRLAM 4D-Var". In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 135.645, pp. 2100–2109.

Stroud, Jonathan R, Matthias Katzfuss, and Christopher K Wikle (2018). "A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation". In: *Monthly Weather Review* 146.1, pp. 373–386.

Teixeira, Joao et al. (2014). "Atmospheric General Circulation Models". In: *Encyclopedia of Remote Sensing*. New York, NY: Springer New York, pp. 35–37. ISBN: 978-0-387-36699-9. DOI: 10.1007/978-0-387-36699-9_8. URL: https://doi.org/10.1007/978-0-387-36699-9_8.

Tippett, Michael K et al. (July 2003). "Ensemble Square Root Filters". In: *Monthly Weather Review* 131.7, pp. 1485–1490. ISSN: 0027–0644. DOI: 10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2.

Tong, Xin T (2018). "Performance analysis of local ensemble Kalman filter". In: *Journal of Nonlinear Science* 28.4, pp. 1397–1442.

Verstraete, Gylian, El-Houssaine Aghezzaf, and Bram Desmet (Jan. 2020). "A leading macroeconomic indicators' based framework to automatically generate tactical sales forecasts". In: *Computers & Industrial Engineering* 139, p. 106169. ISSN: 0360-8352. DOI: 10.1016/J.CIE.2019.106169. URL: https://www.sciencedirect.com/science/article/pii/S0360835219306382.

Vetra-Carvalho, Sanita et al. (2018). "State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems". In: *Tellus A: Dynamic Meteorology and Oceanography* 70.1, pp. 1–43. DOI: 10.1080/16000870.2018.1445364. URL: https://doi.org/10.1080/16000870.2018.1445364.

Wallace, John M and Peter V Hobbs (2006). *Atmospheric science: an introductory survey*. Vol. 92. Elsevier.

Wang, Xuguang et al. (2007). "A comparison of hybrid ensemble transform Kalman filter–optimum interpolation and ensemble square root filter analysis schemes". In: *Monthly weather review* 135.3, pp. 1055–1076.

Westgate, Philip M (2016). "A covariance correction that accounts for correlation estimation to improve finite-sample inference with generalized estimating equations: a study on its applicability with structured correlation matrices". In: *Journal of statistical computation and simulation* 86.10, pp. 1891–1900.

Wikle, Christopher K and L Mark Berliner (2007). "A Bayesian tutorial for data assimilation". In: *Physica D: Nonlinear Phenomena* 230.1-2, pp. 1–16.