Thesis Unofficial Clickable Version: Multiple Frequency Particle Filtering Approaches in Paleoclimatology

January 16, 2022

Acknowledgements

I would like to express great thanks to my supervisors, Pierre-Antoine Absil and Hugues Goosse, for their kind patience and support in guiding me through this task. This gratitude also extends to François Klein who helped me greatly until the end. This work would also not have been the same without the continuous support of my parents and my aunt, Dominique, who helped me muster the courage when I needed it most.

Antoine Gilliard

Contents

0.1 Introduction

In the context of the current climate urgency, future climate evolution has become a topic almost everyone on Earth hears, reads or talks about on a daily basis. As learning from the past is often the best way to seek advice for the future, paleoclimatology plays a key role in the tremendous effort of the scientific community to improve long term climate predictions. Combining geological records with computational models, past climates reconstruction is an essential, though arduous and still perfectible task.

Available information on the state of past climates is given by climate proxies. These are climate archives found in nature (corals, tree rings, etc...) which give information on one or more state variables of the climate at a given time [2, 19, 20]. They can stand for direct measurements in the context of climate reconstruction [2].

Climate models tend to be characterised by extremely large state spaces, with millions of dimensions or even more [4, 11]. On top of that, they are also nonlinear [4, 11]. While smoothing filtering methods have long been used to reconstruct the dynamic as precisely as possible [5], recursive mathematical system filtering methods can also come as a powerful tool because they are less costly computationally and offer convincing results [4, 6]. Among those filters, particle filters are especially suited to this task [4]. Their main advantage is that they handle nonlinearity whereas other filters (like the traditional Kalman filter) cannot [7, 8].

Hence, the field of paleoclimatology makes use of data assimilation methods to reconstruct climate states as precisely and efficiently as possible [1, 14]. Proxy information left in the geological record unveils insights about the state of the climate systems over certain ranges of time. This coupled with existing climate models allows us to refine existing information as well as unravel new information.

This document will address one of the main challenges that particle filters commonly face in paleoclimatology: proxy resolution on different timescales. Indeed, proxy records come with a wide variety of time resolutions [2]. Standard particle filters work on observations on a single timescale. Combining information on averages ranging over different timescales is no common task for them [14].

This document will seek to determine how to reconstruct the state of a system when observations are made on two variables which are available over two different timescales. For example, one variable could be the average yearly surface temperature at a location while the second variable could be the average decadal sea surface temperature at another location. For the sake of simplicity, we will systematically refer to these variables by x and y. For this purpose, four different particle filtering methods will be explored.
Our exploration will be structured into three subsections:

1. Introduction on particle filters in paleoclimatology, followed by a theoretical exploration of the four methods.
2. Measure of performance of the four methods on a simple chaotic model. This will be done on Lorenz63 [24].
3. Evaluating the scalability of two out of the four methods by performing assimilation on an intermediate complexity climate model, LOVECLIM [3].

In the second and third sections, the objective will be to evaluate the accuracy of the methods in reconstructing an unknown reference run from noisy observations. To do so, we will systematically follow the underneath protocol:

1. Integrate a dynamical system through time in order to get a reference run.
2. From the reference run computed in 1, take averages on x and y over their respective timescales.
3. Add a known error noise to the averages computed on step 2, giving us pseudoproxies.
4. Use the particle filtering methods to reconstruct the reference run from 1 using the noisy pseudoproxies generated in 3.
5. Assess the reconstruction quality from each method.

The design of these methods will also address an important drawback in particle filter reconstructions on climate models, the curse of dimensionality. High dimensional systems (like climate models) tend to go with high degrees of freedom, which usually require an exponential number of particles to reconstruct [4, 11, 9]. This makes the practical use of particle filters on climate models computationally challenging. We are constrained to limit our particle swarm size, which often leads to particle degeneracy [11, 12]. Therefore, the ability of each method to handle degeneracy will be assessed.

0.2 Notations

We will use the following mathematical notations to write down our equations in this document. The major notations are summarised in Table 1.

- For the system and the reference run
  - k: dimension of the system. A system features the interaction between a certain number of variables. This notation gives the number of variables.
  - dt: timestep length. Simulating the dynamics of a system requires a timestep of integration. This variable is the time length of that step.
  - T: total number of timesteps over which the integration is run
  - u: the reference run. It gives the full evolution of the system over T given in $\mathbb{R}^{k\times T}$ depending on a specific starting position.
  - $u_{v,t}$: value of variable v at timestep t in the reference run
  - $S(u_t)$: returns the time derivatives of the variables of the system given the current state u_t.
- For the noise and the proxies
  - Δ_v: period over which the proxy on variable v provides an averaged value (in # of timesteps). This element does not exist for variables for which we extract no information. It is important to note that in this document we consider two proxies, on variables we call x and y respectively. We will take x as the high frequency observation variable and take y as the low frequency observation variable. Furthermore, we will consider Δ_y as a multiple of Δ_x throughout the document. By way of explanation:

$$\Delta_y > \Delta_x \text{ where } \frac{\Delta_y}{\Delta_x} = l \text{ with } l \in \mathbb{N} \tag{1}$$

  - σ_v: the standard deviation of the white noise $\mathcal{N}(0, \sigma_v^2)$ of the proxy on variable v.
  - p_v: general notation for the proxy information on variable v
  - $p_{v,t}$: proxy value for the noisy average for the variable v over the timestep range $[t - \Delta_v \dots t[$ given as:

$$p_{v,t} = \frac{\sum_{\tau=t-\Delta_v}^{t-1} u_{v,\tau}}{\Delta_v} + \mathcal{N}(0, \sigma_v^2) \tag{2}$$

    Note that this notation is only valid for t values which are multiples of Δ_v, since proxy information is given in chunks of Δ_v timesteps.
- For the particles
  - N: number of particles
  - $\psi_{v,t}^i$: i^{th} particle's value for variable v at timestep t
  - $\mu_{v,t}^i$: i^{th} particle's average value for variable v over range $[t - \Delta_v \dots t[$
  - w_t^i: weight of particle i at timestep t
- Notations for the likelihoods and resampling

– $\varphi(\mu, \sigma, x)$: probability of sampling x from the normal distribution $\mathcal{N}(\mu, \sigma^2)$

– r_t^i: resampling coefficient, number of daughter particles of i^{th} particle after timestep t

– ξ: final trajectory reconstruction, given in $\mathbb{R}^{k \times T}$. This reconstruction will be used to evaluate the performance of a filter in a specific setting.

– $\alpha()$: A boolean function on the likelihoods called the criterion function. This function will be explained in the section on conditional resampling 1.3.5.

Symbol	**Meaning**
dt	Timestep length
k	Dimension of the system
N	Number of particles
p	Proxy information
r	Resampling coefficient
$S()$	Time derivatives of the system
T	Total number of timesteps
u	The reference run
w	Weight of a particle
$\alpha()$	Criterion function
Δ	Proxy resolution in # of timesteps
ξ	Final trajectory
σ	Standard deviation
$\varphi()$	Normal distribution pdf
ψ	Particle trajectory

Table 1: Notation Table

Chapter 1

Particle Filtering and Paleoclimatology

1.1 Climate Models and Data Assimilation

1.1.1 Climate Models

Climate models are numerical tools which are used to simulate the interactions of the different components of the climate, namely the oceans, the atmosphere, the land surface, the ice and the biosphere.

They can be of a wide range of complexities, but can generally be ranked based on complexity within three groups (information from [17]):

- Energy Balance Models (EBMs): As suggested by their name, EBMs look at the climate based on earth's incoming and outgoing radiation to determine the earth's heat storage. The most basic EBMs don't even include any spatial dimensions, they are called zero-dimensional models. Others, more complex, may include latitude based simulations.
- Earth Models of Intermediate Complexity (EMICs): Similar to EBMs, these models are heavy simplifications of the climate system. They nevertheless contain elements of earth's geography and use a grid based approach to simulate interactions between the main drivers of the climate by integrating differential equations through time. They are much more computationally costly than EBMs and have many more degrees of freedom.
- General Circulation Models (GCMs): These models are the most precise climate models to date. Their meshes are generally within the precision of 100km and they try to mimic earth's geography as precisely as possible. These models are by far the most computationally costly.

Applications on climate models can also be tested on systems with similar underlying dynamics. The Lorenz63 model (which was originally developed to investigate atmospheric convection [18]), is one example. This model displays a chaotic behaviour while being of a striking mathematical simplicity. Climate systems are also chaotic, and have many more degrees of freedom than Lorenz63. This makes Lorenz63 a useful tool when evaluating the performance of experimental data assimilation methods with little computational requirement.

1.1.2 Data Assimilation

Data assimilation is the task of combining observation with information on the dynamics of a system. In paleoclimatology, one usually combines information from proxies and a climate system. The state of the system before the inclusion of observation, called the "prior", holds all of the information given by the simulation about the state of the system at a certain

point in time. The updated prior, referred to as "posterior", is the best estimate of the state of the model when combining the prior with observations [14].

There exists a wide variety of mathematical methods to reconstruct a posterior. There are also different ways of obtaining a prior, namely online and offline assimilations:

- Online assimilation: Where the prior is obtained through simulation based on the former posterior. It works in a step by step iterative manner.
- Offline assimilation: When the prior is a set of pre-simulated system states which is independent from the former posterior [25]. This method is widely used in paleoclimate reconstruction because it has a significant computational advantage of not performing any simulation, which makes it possible to use readily available simulations from high complexity models [25, 14].

1.1.3 The Problem of Multiple Timescales

Data assimilation, whether online or offline, often requires to be performed at regular intervals. In paleoclimatology unfortunately, we are often faced with proxies at different time resolutions. Some proxies, such as coral reefs [19] and tree ring records [20] can yield information at yearly resolutions. Others, such as deep sea cores, can have time resolutions in the order of magnitude of a century or more [21]. This makes it exceedingly hard to make use of the data at hand to perform the necessary data assimilation. As a result, researchers often resort to losing a part of the information by bringing high frequency proxies to the lower frequencies, or by clumsily interpolating the low frequencies into the high ones [14].

In their paper [14], Nathan Steiger and co-authors use an offline Kalman filter based approach as an attempt to shed light on this issue. Their method consists of assimilating the data at highest resolution first (shortest timescale). Once the data is assimilated, the mean over the next highest resolution (next shortest timescale) is computed and kept, as well as the high resolution fluctuations around this mean. Once the mean has been assimilated (with the proxies at a lower frequency), the deviations are reapplied to it to conserve the previous assimilation (done at higher frequency). This method, which can only be performed offline, highlights how tricky it can be to use proxies at different timescales without forcefully bringing them on the same one.

In this document, we will, with the four particle filtering methods, investigate the feasibility of online methods of multi-timescale assimilation.

1.2 An Introduction on Particle Filters

This section is heavily inspired from [4]. All formulas come from it, with the exception of Formulas 1.4 and 1.8 which are slightly modified versions of formulas from that same paper (respectively formulas (9) and (15)).

1.2.1 What are Particles?

The objective of data assimilation is to reconstruct the state ψ of a system when given sparse and uncertain information (denoted by proxies p). In order to do so, it is useful to know $\mathbf{P}(\psi|p)$ which denotes the probability of ψ being the state vector based on the proxy observation p. Using Bayes formula, this probability can be computed as follows:

$$\mathbf{P}(\psi|p) = \frac{\mathbf{P}(p|\psi)\mathbf{P}(\psi)}{\mathbf{P}(p)} \tag{1.1}$$

In this equation, the values of the terms $\mathbf{P}(p|\psi)$ and $\mathbf{P}(p)$ are calculated based on the probability of observation of the proxy value p in the first place. $\mathbf{P}(p|\psi)$, the probability of observing p given the system state ψ, can easily be calculated if we know the noise placed on each proxy. The noise is usually normally distributed, hence $\mathbf{P}(p|\psi)$ is denoted by a gaussian, formally:

$$\mathbf{P}(p|\psi) = A \cdot \exp\left\{-\frac{1}{2} \cdot [p - H(\psi)]^T R^{-1} [p - H(\psi)]\right\} \tag{1.2}$$

Where $H(\psi)$ is the measurement operator, which is the model equivalent of the observation p, and R is the noise covariance matrix [4]. The general formulation presented above will be reformulated into products of one dimensional gaussian sample probability throughout the document in order to facilitate the visualisation of the steps used in the methods.

The denominator in Bayes' formula, here $\mathbf{P}(p)$, can be seen as a normalisation term. It can be computed using the following formula:

$$\mathbf{P}(p) = \int \mathbf{P}(p|\psi)\mathbf{P}(\psi)d\psi \tag{1.3}$$

Filters considered in this document are recursive filters, for which we have a prior estimation for our system state which is updated using the noisy observation, giving us a posterior. The posterior is then updated to the prior for the next assimilation step using the system's dynamic.

Particle filters are Monte-Carlo filtering methods for which the prior probability density function (pdf) of the model is approximated by a discrete set of ensemble members also called "particles". Using this discrete distribution, the probability $\mathbf{P}_t(\psi)$ of the state vector being ψ at timestep t is given by:

$$\mathbf{P}_t(\psi) = \frac{1}{N}\sum_{i=0}^{N}\delta(\psi - \psi_t^i) \tag{1.4}$$

Where ψ_t^i denotes the state vector of each independent particle i at timestep t. This formulation of the pdf allows us to make unbiased estimates on the true state of the system. For instance, we can try to estimate ψ_t, the true state of the system variables at timestep t using an arithmetic mean as follows:

$$\psi_t \approx \frac{1}{N}\sum_{i=0}^{N}\psi_t^i \tag{1.5}$$

This estimate is by definition unbiased since it is obtained from taking an overall mean of a sample.

1.2.2 Sequential Importance Sampling Filter (SIS)

The most basic sampling filter is the Sequential Importance Sampling Filter. Importance sampling is a method in statistics used to determine properties of a certain distribution with a sample set of another distribution. This can be done when one has an additional distribution which helps in describing the target distribution.

This fits with the nature of particle filtering problems. The distribution of interest is the possible state of the system in the reference run. The sample set of a different distribution corresponds to the propagated particles. Finally, the additional information comes from known noisy observations of the system. This allows to calculate weights for importance sampling on the particles.

This approach makes use of the Bayes formula (Equation 1.1) to compute the likelihood associated with each particle. Equation 1.4 gives the discrete distribution prior to likelihood calculation. Holding proxy information p, the distribution can be updated as follows:

$$\mathbf{P}_t(\psi|p) = \sum_{i=0}^{N} w_i\delta\left(\psi - \psi_t^i\right) \tag{1.6}$$

Where all the weights w_i sum up to 1, and are denoted by:

$$w_i = \frac{\mathbf{P}_t\left(p|\psi_t^i\right)}{\sum_{j=0}^{N}\mathbf{P}_t\left(p|\psi_t^j\right)} \tag{1.7}$$

Similarly to Equation 1.5, one can now approximate the true state of the system using a weighted mean over the particles based on their respective likelihoods:

$$\psi_t \approx \sum_{i=0}^{N} w_i \psi_t^i \tag{1.8}$$

Sequential importance sampling is one of the most straightforward ways to apply a particle filter. Starting with an initial distribution one gets sample trajectories of the system and by using observations one gets to reconstruct it as accurately as possible. This method has nevertheless a more significant drawback, especially in chaotic systems, which are representative of geophysical systems. This drawback comes from filter degeneracy. As time moves on, the sample particles tend to deviate from the target trajectory. Hence, one often ends up with a very uneven distribution in the weights, with most particles of the sample being totally unlikely and some heavily weighted ones still being a poor representation of what the dynamic actually is.

There are different ways to try to reduce this variance in the weight distribution among the particles in order to obtain a fuller picture. One of the simplest ones, and the one that will be used in this document, is resampling.

1.2.3 Sequential Importance Resampling Filter (SIR)

SIS filters can be subject to significant degeneracy issues even in low dimensional chaotic systems with very few degrees of freedom. To avoid this, we use a powerful tool called resampling. When simulating for a prior in Sequential Importance Sampling we are often faced with particles that have strayed away from the actual trajectory. These have become of negligible importance to the filter. Hence, it comes as a practical solution to eliminate those low likelihood particles from our sample and "resample" by creating duplicates of particles with high likelihoods in their stead. The basic functioning of a SIR particle filter follows these steps after initialisation:

1. Run the particles, sometimes noisily
2. Compute their likelihoods and their weights
3. Resample the particles, return to step 1.

The pseudocode for this method is displayed in section 1.3.2 in algorithm 1.

After each resampling step, we want our new sample to be a proportional representation of our old sample with respect to their weights. Hence, when looking for the resampling coefficient r_t^i of particle i at timestep t, one can simply multiply the weight of a particle by N. This does not give an integer however, and one cannot simulate a fraction of a particle. The "mother" particles from the old sample, must each have an integer number of "daughter" particles in the new sample. Different methods exist to go around this issue. In this document we used residual resampling. Residual resampling, as well as all unbiased resampling methods respect the following property:

$$\mathbb{E}\left[r_t^i\right] = w_t^i \cdot N \tag{1.9}$$

The differentiation of daughter particles is also relevant. If daughter particles are just pure copies of their mothers in a deterministic system (like much of the climate models), their trajectories are bound to be exactly the same. One of the ways of ensuring that this does not happen is by adding a random perturbation to the dynamic of daughter particles, thus ensuring a differentiation before the next resampling step. This is usually done by adding noise to the dynamic of the system for an arbitrary number of timesteps after each resampling, or by slightly perturbing the state of the daughters particles.

In our new SIR filter, we can get an unbiased estimate of the state of the system by taking a weighted mean of the state of the different particles. Similar to equations 1.8 and 1.5, we

can now write:

$$\psi_t \approx \frac{1}{N} \sum_{1}^{N} r_t^i \cdot \psi_t^i \tag{1.10}$$

SIR filters solve for many degeneracy issues encountered when there is no resampling. Still, it does not solve for all degeneracy issues. Divergence can still become a problem when the number of particles is too small and the number of degrees of freedom of the system is too large. Usually, degeneracy happens when there are too many timesteps between observations. Since we cannot perform a resampling without observation, the time elapsed between resampling steps can become too long to avoid degeneracy.

1.3 Investigated Multi-timescale Particle Filtering Methods

Throughout this section, the particle filtering methods investigated in this document will be described. We will seek to determine their advantages and drawbacks in order to make predictions on their potential usefulness. In order to illustrate the behaviour of the different methods, we will use numberless visual examples. It is important to note that in these examples, for visualisation purposes, proxy information seems to be given at regular intervals on punctual expected state of the system (e.g. noisy observation on the state of variable x at timestep n). In the actual filters we are interested in the average state of the system over a certain number of timesteps (e.g. noisy observation on the average state of variable x in timestep range $m...n$).

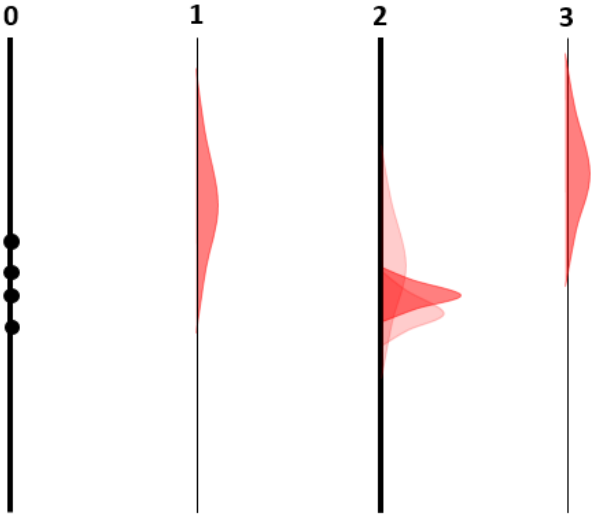

Figure 1.1: Illustration of data given over the different timesteps

Example on information use: 4 particles are used, high frequency proxies are available at time 1,2,3, low frequency proxies are available at time 2. As can be seen on observation 2, the combined interpretation of the high frequency proxy and the low frequency proxy can give another probability distribution corresponding to the multiplication of the two.

The methods described in this section are summarised in Table 1.1, it can be used as a tool to follow the explanations throughout the reading of this part of the document.

1.3.1 Common Steps of the Four Particle Filtering Methods

Each particle filter described below is different. They nevertheless share overarching mechanisms which need to be formalised beforehand.

Propagating Particles

Particles ψ_i are propagated through time using an integration scheme. The propagation (using an Euler explicit integration scheme) is as follows:

$$\psi_{t+1}^i = \psi_t^i + S(\psi_i^t) \cdot dt \tag{1.11}$$

Noise can be added as a way to induce differentiation on daughter particles:

$$\psi_{t+1}^i = \psi_t^i + S(\psi_i^t) \cdot dt + \mathcal{N}(0, \Sigma) \tag{1.12}$$

Where Σ is a diagonal correlation matrix; its diagonal terms are typically less than $\frac{\sigma_v^2}{\Delta_v}$.

It is important to note that Formulas 1.11 and 1.12 are applicable on assimilation on Lorenz63 in particular, where both the propagation and differentiation of the particles will work exactly in this fashion. We will not influence the dynamic of the system in LOVECLIM. For the propagation of the particles, we will follow the integration scheme imposed by the model. For the differentiation of the particles, we will add a perturbation to the state of daughter particles before feeding them back to the model.

Computing the likelihoods and the weights

Computing the likelihood associated to each particle at a certain timestep requires different information. It requires to know exactly the state of the particle, the noisy observation at that timestep and the distribution of the noise on this observation. In this section, we assume the noise to be normally distributed. The way of calculating $\mathbf{P}_t\left(p|\psi_t^i\right)$, will vary from one method to the other.

The weights computation is always the same and consists of a scaling of likelihoods with respect to the sum of all likelihoods(as shown in Equation 1.7).

Residual resampling

The way particles are resampled is uniquely dependent on their weight. The resampling scheme used in this document is called residual resampling. Resampling is about making the next particle swarm a proportional representation of the previous swarm according to their respective weights (i.e. the particle i should be reproduced $w_i N$ times). Residual resampling happens in two steps. In the first step, every particle with weight superior to $\frac{1}{N}$ (which has a resampling coefficient greater than 1) is resampled with the integer part of its resampling coefficient [22]. In the second stage, the remaining particles are elected at random proportionally to the remaining portion of their respective weights, called the residues [22]. For example, a particle with exact resampling coefficient 4.63 is resampled 4 times. The residue of that particle is 0.63.

Hence, we have:

$$r_t^i = \mathbf{Pt_1}\left(w_t^i\right) + \mathbf{Pt_2}\left(w_t^i\right) \tag{1.13}$$

Where $\mathbf{Pt_1}\left(w_t^i\right)$ is the first part of the resampling procedure, and $\mathbf{Pt_2}\left(w_t^i\right)$ the second part. Where:

$$\mathbf{Pt_1}\left(w_t^i\right) = \left\lfloor w_t^i \cdot N \right\rfloor$$

and $\mathbf{Pt_2}\left(w_t^i\right)$ is defined such that:

$$\mathbb{E}\left[r_t^i\right] = w_t^i \cdot N \text{ and } \sum_{i=0}^{N-1} r_t^i = N$$

1.3.2 First Method: Base Method (Timescale Selection)

This first method is the regular SIR particle filter (Section 1.2.3). This filter works on a single timescale, which will require a selection between the different timescales at hand.

One can use the proxy information in two different ways with this algorithm. As a first approach, it is reasonable to disregard one of the two proxies and assimilate everything on its appropriate timescale, unavoidably getting rid of part of the available information. This method will be called the Base Method without interpolation. As a second approach, we could try to interpolate the low frequency proxy, transforming it to make it match with the high frequency one, thus ending up with two different observations at each assimilation. This method will be called the Base Method with interpolation.

The pseudocode of this method is given as:

Initialise the state of the particles at timestep $t = 0$;
while $t<T$ **do**
    propagate particles in time noisily (1.12);
    $t = t + 1$;
    if $t \mod \Delta_x = 0$ **then**
        compute likelihoods of each particle (1.14 or 1.15);
        calculate the weights (1.7);
        resample the particles (1.13);
    end
end

Algorithm 1: Base Method, an application of a SIR filter

Every step of this algorithm is elaborated upon in the following sections.

Computing the likelihoods

Since we can assimilate with or without interpolation, there can be two ways of computing the likelihoods on each particle before calculating the weights. In the Base method without interpolation, the probability is obtained in the following way:

$$\mathbf{P}\left(p_x|\psi_t^i\right) = \varphi\left(\mu_{x,t}^i,\, \sigma_y,\, p_{x,t}\right) \tag{1.14}$$

As a reminder, $\varphi\left(\mu,\, \sigma,\, x\right)$ gives the probability of sampling x from the normal distribution $\mathcal{N}(\mu, \sigma^2)$.

In the Base Method with interpolation, we linearly interpolate y to the timescale of x observations. Doing so, we can use additional information when computing the likelihood. We will call y' the interpolated y proxy on x timescale. We have:

$$\Delta_{y'} = \Delta_x$$

The likelihood computation becomes:

$$\mathbf{P}\left(p_x, p_{y'}|\psi_t^i\right) = \varphi\left(\mu_{x,t}^i,\, \sigma_x,\, p_{x,t}\right) \cdot \varphi\left(\mu_{y',t}^i,\, \sigma_{y'},\, p_{y',t}\right) \tag{1.15}$$

The fact that the distribution of the error is uncorrelated between the different proxies allows to calculate the overall likelihood by just multiplying the two together.

Comments on the method

This first method, which is a regular SIR particle filter, offers the simplest approach to the problem. It will serve as a benchmark to assess the performance of the subsequent particle filters explored in this document.

The two ways of doing this method (i.e. with and without interpolation of the y proxies) will be compared as well. Not interpolating the y proxies makes the assimilation unbiased and straightforward, but some information is lost. Using interpolation on y proxies on the other hand is a delicate procedure. While it can bring more precision if y averages are interpolated accurately, it can also divert the particles into unwanted territory if the interpolation is inaccurate.

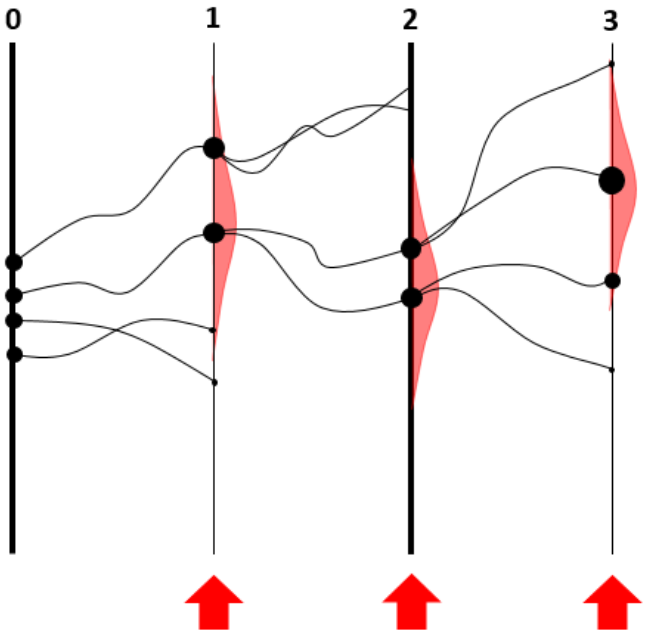

Figure 1.2: Illustration of the Base Method without interpolation

Illustration of the functioning of the Base Method without interpolation, example with 4 particles. In this case, we consider the high frequency proxies and ignore the low frequency proxies to make our data assimilation, thus without interpolation. Resampling steps are made on every observation.

1.3.3 Second Method: Particle Backtracking

The Particle Backtracking Method works in a similar fashion than the Base Method, resampling at every timestep at which information is available. The main difference is that it takes into account the low frequency information as it is when resampling. We now resample differently based on our different timescales Δ_x and Δ_y. This method has been used with LOVECLIM. However, results are not yet available.

The pseudocode of this method is given as:

Initialise the state of the particles at timestep $t = 0$;
while $t<T$ **do**
  propagate particles noisily (1.12);
  $t = t + 1$;
  if $t \mod \Delta_x = 0$ **then**
    if $t \mod \Delta_y = 0$ **then**
      compute likelihoods of each particle (1.16);
    end
    else
      compute likelihoods of each particle (1.14);
    end
    calculate the weights (1.7);
    resample the particles (1.13);
  end
end

Algorithm 2: Particle Backtracking

Computing the likelihoods

The particle's likelihoods are calculated differently if t is a multiple of Δ_y or not. Likelihoods in this algorithm are computed every $\Delta_x dt$ (e.g. whenever t is a multiple of Δ_x). When t is a multiple of Δ_x but not of Δ_y, likelihoods are calculated in the same fashion than in 1.14. However, whenever we can assimilate on p_y (when t is a multiple of Δ_y), we calculate the likelihoods of the particles using both p_x and p_y:

$$\mathbf{P}\left(p_x, p_y | \psi_t^i\right) = \varphi\left(\mu_{x,t}^i,\, \sigma_x,\, p_{x,t}\right) \cdot \varphi\left(\mu_{y,t}^i,\, \sigma_y,\, p_{y,t}\right) \tag{1.16}$$

This likelihood calculation couples the high frequency (p_x) observation with the low frequency (p_y) observation. It is important to note that the averages over Δ_y are made by conserving the genealogy of each particle. This is necessary since multiple resamples based on p_x can occur in each Δ_y chunk. This is why we decided to call this method Particle Backtracking. In order to better understand how the particle genealogy is conserved, additional information can be found in Appendix.1.

Comments on the method

The idea behind this method is that particles are directed using high frequency resamplings in between low frequency resamplings. The discrete swarm of particles obtained from high frequency resampling is an unbiased reflection of the probability distribution of the trajectory as given in 1.6. Further modification of this distribution using the particle genealogy (as explained in Appendix.1) can also be considered as an update of the probability distribution based on future given information.

This method is advantageous because it takes all the information given into account to reconstruct the state of the system. However, this method can lead to degeneracy if we end up with only one relevant continuous trajectory at a time between resamplings at a low frequency (Δ_y).

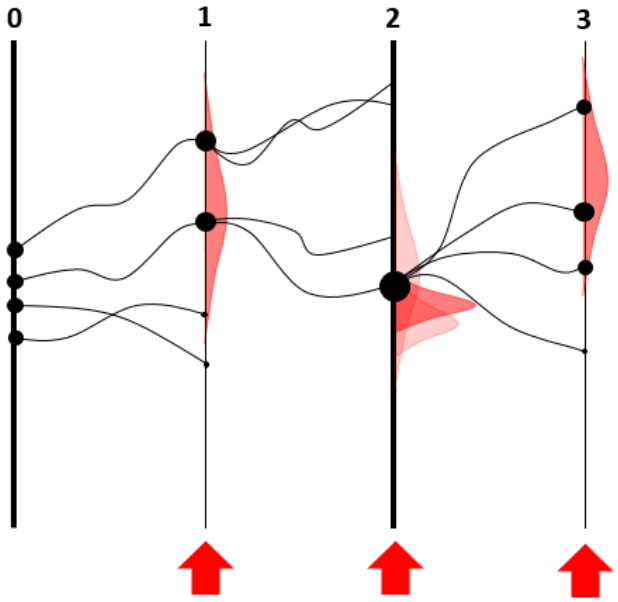

Figure 1.3: Illustration of the Particle Backtracking Method

Example of Particle Backtracking: 4 particles are used. Resampling is made on every observation and the probability distribution used on even timesteps corresponds to the combined proxy probability distribution.

1.3.4 Third Method: Cumulative Resampling

The Cumulative Resampling Method resamples at the lowest frequency only (here on y). It multiplies likelihoods at high frequency together in between low frequency resamples in order to take it into account in the final reconstruction. It functions as a state vector augmentation [23], taking multiple observations in the state vector to estimate a more accurate system trajectory. The pseudocode of this method is given as:

Initialise the state of the particles at timestep $t = 0$;
while $t<T$ **do**
  if $t \mod \Delta_y < \Delta_x$ **then**
    propagate particles in time noisily (1.12);
  end
  else
    propagate particles in time without noise added (1.11);
  end
  $t = t + 1$;
  if $t \mod \Delta_y = 0$ **then**
    compute likelihoods of each particle (1.17);
    calculate the weights (1.7);
    resample the particles (1.13);
  end
end

Algorithm 3: Cumulative Resampling

Computing the likelihoods

In this specific case, the particles are projected and run over Δ_x before they are to be resampled. Hence, their respective likelihoods are the result of the multiplication of likelihoods in

each segment Δ_y as well as on the segment Δ_x. The calculation is given as:

$$\mathbf{P}\left(p_x, p_y | \psi_t^i\right) = \varphi\left(\mu_{y,t}^i,\, \sigma_y,\, p_{y,t}\right) \cdot \prod_{i=0}^{l-1} \varphi\left(\mu_{x,(t-i\Delta_x)}^i,\, \sigma_x,\, p_{x,(t-i\Delta_x)}\right) \tag{1.17}$$

Comments on the method

This method allows to combine independent probabilities over multiple timesteps to better evaluate particle likelihoods. It is the most rigorous application of an SIR particle filter using information given at high and at low frequency.

Nevertheless, despite it's strong theoretical basis, it can have some important flaws, especially in chaotic systems. This method has the most timesteps (therefore most degrees of freedom) in between resamplings. This comes with a high risk of degeneracy if the number of particles is too low and if the lowest frequency assimilation timescale is too long. We can easily end up with a particle swarm with very few particles or no particle at all correctly depicting the actual trajectory of our system.

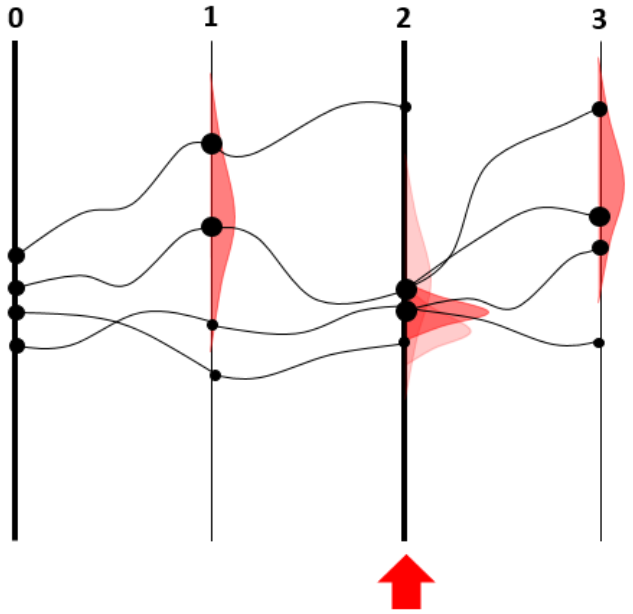

Figure 1.4: Illustration of the Cumulative Resampling Method

Example of Cumulative Resampling: 4 particles are used. Resampling is made on every even observation (low frequency), and the probability distribution used on even timesteps corresponds to the combined proxy probability distribution.

1.3.5 Fourth Method: Conditional Resampling

This type of resampling can be seen as a mix between Particle Backtracking and Cumulative Resampling. To compensate for the risk of degeneracy in the Cumulative Resampling, we impose a criterion on the likelihoods of the different particles such that, if this criterion is met, we resample them at high frequency, much like the Particle Backtracking Method.

Degeneracy conditions can take many types of criteria into account. The choice of criterion can be established empirically based on the evaluated performance of each criterion on a particular system. In this document, two criteria were evaluated on Lorenz63, entropy

(Section 2.5) and the Gini coefficient (Appendix .2). Only entropy was eventually selected as a possible interesting criterion. The Conditional Resampling Method was not used with LOVECLIM.

The pseudocode of this method is given as:

Initialise the state of the particles at timestep t=0;
resampled=0 (gives the timestep of last resampling);
while $t<T$ **do**
  if $resampled - t < \Delta_x$ **then**
    propagate particles in time noisily (1.12);
  end
  else
    propagate particles in time without noise (1.11);
  end
  $t = t + 1$;
  if $t \mod \Delta_x = 0$ *and* $t \mod \Delta_y \neq 0$ **then**
    compute likelihoods of each particle (1.14);
    cumulate likelihoods (1.18);
    if $\alpha(L)$ (Subsection criterion function in 1.3.5) **then**
      calculate the weights (1.7);
      resample the particles (1.13);
      reset likelihoods(1.19);
      resampled=t;
    end
  end
  if $t \mod \Delta_y = 0$ **then**
    compute likelihoods of each particle (1.16);
    cumulate likelihoods (1.18);
    calculate the weights (1.7);
    resample the particles (1.13);
    reset likelihoods(1.19);
    resampled=t;
  end
end

Algorithm 4: Conditional Resampling

The likelihood vector L

This method sees the introduction of a likelihoods vector L_t, for which t denotes the time of last observation. One can do two things when it comes to using the likelihood vector:

- Cumulating likelihoods: This is done when particle likelihoods are updated based on a recent observation. We have to multiply the likelihood of each current particle by the likelihood it had before. This updates the likelihood vector L as follows:

$$L_{t+\Delta_x} = L_t \circ \left[\mathbf{P}\left(p|\psi_t^0\right) \quad \mathbf{P}\left(p|\psi_t^1\right) \ldots \mathbf{P}\left(p|\psi_t^N\right)\right] \tag{1.18}$$

  Where $\circ$ denotes the elementwise Hadamard product between the two vectors.

- Resetting likelihoods: This is done after a resampling step in which particle likelihoods have been used in weights to make a new sample. The new particles need to have the same likelihoods. Resetting the likelihoods is setting all particle likelihoods to 1 in L.

$$L_{t+\Delta_x} = [1 \quad 1 \ldots 1] \tag{1.19}$$

The criterion function $\alpha(L)$

The criterion function is at the heart of this particular resampling method. This function determines when to resample and when not to resample in order to have the best possible

reconstruction of trajectory. It is a function of the likelihoods of the particles and can be of any sort that suits best depending on the problem. We will call it $\alpha(L)$ where L is the likelihood vector.

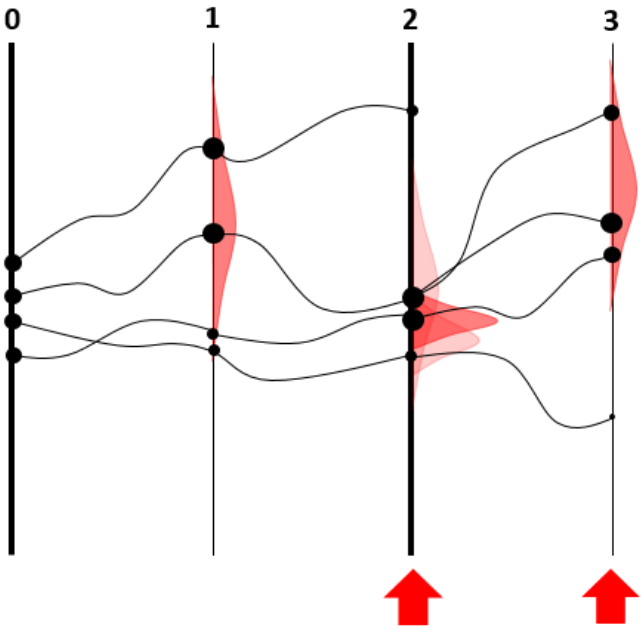

Figure 1.5: Illustration of the Conditional Resampling Method

Example of Conditional Resampling: 4 particles are used. Resampling is automatically made on every even observation (low frequency) and the probability distribution used on even timesteps corresponds to the combined proxy probability distribution. However, as can be observed on timestep 3, the criterion on the likelihoods is met (bottom particle is out of state space) and a resampling is made over the set of particles.

1.3.6 Summary Table of the Methods

In order to illustrate the way the different methods function, the following table can be used as a recapitulation:

Method	**Based off of p_y**	**Resampling**	**Cumulates likelihoods**	**Backtracking**
Base	No	Every $\Delta_x dt$	No	No
Base-Interpolation	Yes, interpolated	Every $\Delta_x dt$	No	No
Backtracking	Yes	Every $\Delta_x dt$	No	Yes
Cumulative	Yes	Every $\Delta_y dt$	Yes	No
Conditional	Yes	When necessary and every $\Delta_y dt$	Yes	Yes

Table 1.1: Summary of the different methods and their characteristics

Chapter 2

Applications to Lorenz63

2.1 The Lorenz63 System

Originally designed to model atmospheric convection [18], the Lorenz63 ordinary differential equation system serves as a good exploration tool when it comes to testing data assimilation methods in paleoclimatology. Firstly, its simplicity makes its analysis very accessible. Secondly, it is nonlinear and chaotic. The system is the following:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x) \tag{2.1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y \tag{2.2}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z \tag{2.3}$$

The σ, ρ and β are constants which impact the dynamic of the system. The standard values for these parameters are $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$. These will be the values used throughout this section. It is of little relevance to our experiment to try for different values of these parameters.

It can be computed that the system has three equilibria by solving the system for $(\dot{x}, \dot{y}, \dot{z}) = (0, 0, 0)$. We get the fixed equilibrium (eq) points:

$$\mathrm{eq}_1 = (0, 0, 0) \text{ and } \mathrm{eq}_{2,3} = \left(\pm\sqrt{\beta \cdot (\rho - 1)}, \pm\sqrt{\beta \cdot (\rho - 1)}, \rho - 1\right) \tag{2.4}$$

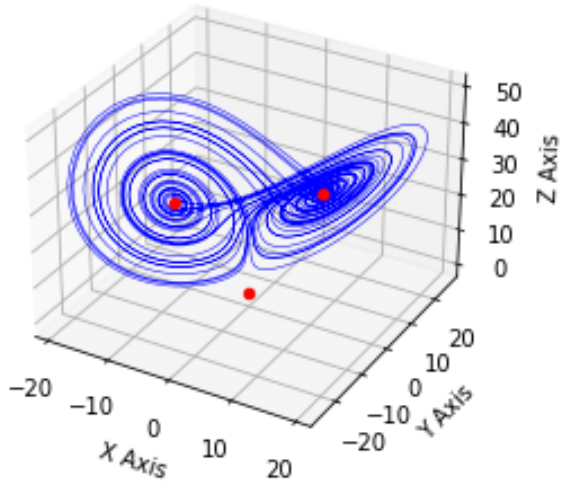

Figure 2.1: Numerical Integration of Lorenz63 with equilibria

In his paper [24], Lorenz analyses the dynamics of this particular system in detail. He observes that above a certain threshold of the maximum z value in a cycle, the system switches

lobes. This observation explains why particle filters can easily degenerate on Lorenz63. Given our noisy observations, if most particles are mistakenly redirected below or above this cutoff value when the true trajectory does otherwise, the filter is bound to face degeneracy. Following the system dynamics therefore becomes particularly touchy when a bifurcation happens [10].

2.2 The Experimental Setup

In this section, we will seek to evaluate how effective the investigated different methods are in reconstructing a reference trajectory based on noisy observation given at different timescales. The reference run uses an Euler integration scheme (with $dt = 0.01$).

2.2.1 Noise

This experiment consists in observing how the particle swarm responds to the noisy observations of the reference trajectory. In order to mirror paleoclimatology proxies, the proxy observations in this experiment will be averages over a certain number of timesteps on x and on y (Δ_x and Δ_y). The first step consists of adding the noise to the reference simulation by adding a white gaussian noise to each point in the trajectory.

Throughout these experiments, the noise added to each point of the reference trajectory will be a tridimensional white noise of $\mathcal{N}\left(0, \sigma_n^2 \cdot I\right)$. The proxies p_x and p_y are averages over this noisy trajectory. The parameters of the noise over these averages $\mathcal{N}\left(\mu_v, \sigma_v^2\right)$ (where v can be x or y) can be computed:

$$\begin{aligned}\mu_v &= \mathbb{E}\left[\frac{\sum_1^{\Delta_v} \mathcal{N}(0, \sigma_n^2)}{\Delta_v}\right] \\ &= \frac{\sum_1^{\Delta_v} \mathbb{E}\left[\mathcal{N}(0, \sigma_n^2)\right]}{\Delta_v} \\ &= \frac{\sum_1^{\Delta_v} 0}{\Delta_v} = 0\end{aligned}$$

and

$$\begin{aligned}\sigma_v^2 &= \mathbf{Var}\left[\frac{\sum_1^{\Delta_v} \mathcal{N}(0, \sigma_n^2)}{\Delta_v}\right] \\ &= \frac{1}{\Delta_v^2}\mathbf{Var}\left[\sum_1^{\Delta_v} \mathcal{N}(0, \sigma_n^2)\right] \\ &= \frac{1}{\Delta_v^2} \cdot (\Delta_v \cdot \sigma_n^2) = \frac{\sigma_n^2}{\Delta_v}\end{aligned}$$

This method makes the proxy noise variance inversely proportional to the number of timesteps, hence making the low frequency proxies contain more information than the high frequency ones. This unbalance is counteracted by the fact that high frequency proxies are used more often. Furthermore, it can easily be observed that by averaging the high frequency proxies over the timescale of the low frequency ones, we have the same noise, making the high frequency proxies at least as relevant as the low frequency proxies when it comes to held information.

The value of σ_n used throughout the Lorenz63 experiment will be $2\sqrt{10}$. This is such that the variance of the error on a proxy with timescale resolution $10dt$ is 4.

2.2.2 Multivariate Analysis

This experiment will see a change in different parameters to evaluate the performance of each method in different cases. This analysis will have 200 runs on each parameter setup for each particle filtering method in order to have an accurate idea on the performance of our assimilation.

Parameter 1: Scale of simulation

We want to assess whether the investigated methods are scalable to EMICs, which contain many more degrees of freedom than the Lorenz63 system. In order to evaluate this scalability, we will perform reconstructions over different scales.

- The first scale is where the high frequency proxy (on x) is an average over 10 timesteps ($\Delta_x = 10dt$). This is a small portion of a cycle in Lorenz63.
- The second scale is a scale with high frequency proxy having averages over 100 timesteps ($\Delta_x = 100dt$). Hence, it has more degrees of freedom and is increasingly subject to the chaos of the system.

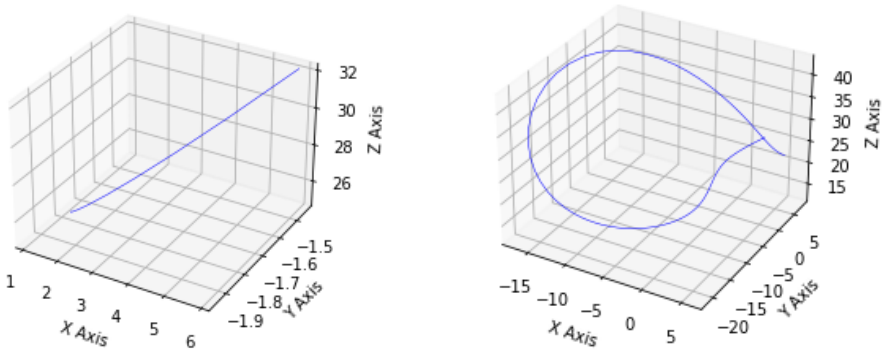

Figure 2.2: Lorenz63 trajectory visualisation $10dt$ and $100dt$

These two plots feature parts of an arbitrary trajectory of the Lorenz63 system with $dt = 0.01$. On the left, the trajectory is given over 10 timesteps, it is only a small part of a cycle. On the right, it is given over 100 timesteps, it is big part of a cycle. It helps in visualising how different it can be to perform an assimilation on these two timescales.

In order to have an equal number of resamplings per total assimilation, the length of simulation will be increased from 2400 timesteps (for $\Delta_x = 10dt$) to 24000 timesteps (for $\Delta_x = 100dt$).

Parameter 2: Ratio between timescales

Following Equation 1, we have that Δ_y is a multiple of Δ_x. In order to evaluate the performance of each method given different situations, we will seek to observe how the methods perform by changing the factor between Δ_x and Δ_y. Checking for:

$$\Delta_y = \Delta_x \cdot \{2, 3, 4, 6, 8, 10\} \tag{2.5}$$

Parameter 3: Number of particles

Observing the behaviour of a method based on the number of particles is a fundamental experiment in particle filtering. All particle filters are supposed to get better and better as the number of particles increases. We will test for the following values of N:

$$N = \{10, 18, 30, 56, 100, 180, 300, 560, 1000, 1800, 3000\} \tag{2.6}$$

This follows an exponential progression, which helps in evaluating the asymptotic performance of the filter. An important aspect to note is that, in our setup, the particles are initialised in the vicinity of the origin of the true trajectory. Most of the filters behave correctly regardless of the origin, but this makes for a more consistent reconstruction.

2.2.3 Performance Measures

Much like in many papers in paleoclimatology [13, 25] (and as seen previously in Equation 1.10), it is possible to reconstruct the trajectory of the system by taking the average of the

trajectories of the particles over chunks of time across which they are weighted by their respective resampling coefficients (on that specific proxy chunk).

$$\xi_{v,t} = \frac{1}{N} \sum_{1}^{N} r_t^i \cdot \psi_{v,t}^i \tag{2.7}$$

We can measure the performance of a method by comparing the true path of the system with that reconstructed trajectory. Although not used in this document, the measurement of the spread of the swarm gives an idea of the uncertainty on this trajectory reconstruction.

Root Mean Square Error (RMSE)

We will evaluate the quality of the reconstructions using a very common metric, the RMSE. The RMSE applied to the Lorenz63 system consists in taking the Root Mean Square of the euclidean distance between the reconstructed trajectory and the true trajectory. Hence:

$$RMSE = \sqrt{\frac{\sum_{t=0}^{T-1} \sum_{v \in \{x,y,z\}} (u_{v,t} - \xi_{v,t})^2}{T}} \tag{2.8}$$

We can take as benchmark the RMSE of a "free" trajectory, which consists of a particle reconstruction trajectory without any assimilation. This trajectory's RMSE averages around 15 [26].

Pearson correlation coefficient (ρ)

Measuring the performance of particle filters on multidimensional systems can also be done with correlation coefficients. This metric will be less solicited to our performance assessment in this chapter of the document. Nevertheless, we felt the need to introduce it here since it will be predominantly used in the next chapter using the LOVECLIM model.

The Pearson correlation coefficient measures the strength of the linear relationship between two variables. It evolves between -1 and 1, with negative correlation meaning the variables evolve oppositely while a positive correlation means the series evolve together. The correlation coefficient between series X and X' is given by 2.9.

$$\rho_{X,X'} = \frac{\text{cov}(X, X')}{\sigma_X \sigma_{X'}} \tag{2.9}$$

This metric will allow to evaluate how well the trajectory of the reconstruction (ξ) follows the trajectory of the reference run (u) on each variable of the system (x, y and z).

2.3 RMSE Results

2.3.1 Base Method

The Base Method (method of timescale selection) consists in a very basic SIR filter, it has been done in the past with different setups [4][10]. The SIR filter is shown to have a decent effectiveness in approaching the true value of observation [10]. In this section, we observe its behaviour with our parameters and setup.

Without Interpolation

Without interpolating, we get an average reconstruction which is, as one might expect, more accurate with an increasing number of particles.

The quality of reconstruction seems to plateau around 3 and 7.5 respectively on the left and right graphs in Figure 2.3. This implies that once this value of RMSE is reached, the quality of reconstruction cannot increase significantly with the number of particles. This threshold is

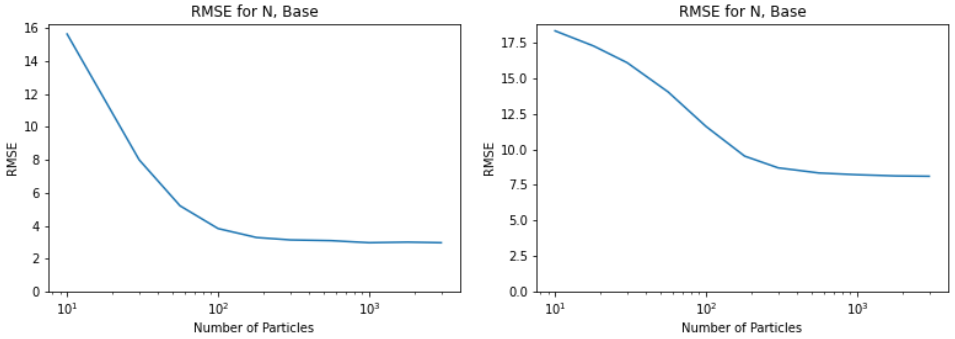

Figure 2.3: Base Method without interpolation $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots made with the Base Method give the value of RMSE versus the number of particles in the simulation. Since it is without interpolation, only the observations on x are used in this filter. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right.

reached at around 100 particles for the short timescales ($\Delta_x = 10dt$) and around 1000 for the long timescales ($\Delta_x = 100dt$). The reconstruction is systematically worse for long timescale reconstructions than for short ones. Indeed, since the observation is more sparse with averages over long timescales, the reconstruction is bound to be worse than on short timescales, where the trajectory can be closely followed more easily. It is important to note that in such reconstructions, the RMSE is heavily skewed by extreme values when particles have diverged.

With Interpolation

Interpolating y proxies can be a delicate procedure. In this experiment, we decided to interpolate them and consider that their observation noise is equal to that on x. The assimilation was thereafter done by taking into account both likelihoods on x values and on y' values.

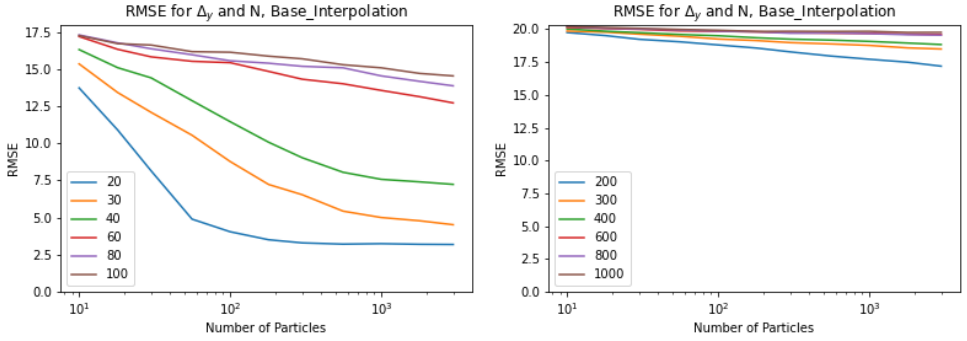

Figure 2.4: Base Method with interpolation $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots made with the Base Method with interpolation give the value of RMSE versus the number of particles in the simulation. Observations on x and observations on y (interpolated) are used in this filter. Each line color is used for a different y proxy timescale (as given before interpolation). High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right.

In Figure 2.4 the assimilation with $\Delta_x = 10dt$ only seems to work for $\Delta_y = 20dt$. Past $\Delta_y = 20dt$ and $\Delta_x = 10dt$, the interpolation does not bring interesting results, this is especially visible in the case where $\Delta_x = 100dt$. This likely comes from the fact that the evolution from one y proxy to the next is highly nonlinear over longer timescales (see Figure 2.2). Interpolating for y in these cases gives a very wrong depiction of the actual proxy data.

This observation highlights the fact that it is very important to know the nature and dynamic of a certain variable before we start interpolating it.

Comparison

The Base Method with interpolation provides similar results for $\Delta_y = 20dt$ than the Base Method without interpolation. Indeed, when measured for 10 particles we have RMSE values of 13.7 with interpolation (Figure 2.4) versus 15.7 without (Figure 2.3), evolving to the values of 3.2 and 3.0 respectively with 1000 particles. However, for any other setup, the Base Method with interpolation works systematically worse than without interpolation.

Overall, it comes as no surprise that taking only one of the two variables comes as the lesser of two evils. Thus, we will use the base method without interpolation as a benchmark.

2.3.2 Particle Backtracking

The second method to be tested is the Particle Backtracking Method. The assimilation in this method is performed very similarly than in the Base Method without interpolation. The low frequency information is still used in the likelihood calculation at the end of each Δ_y chunk.

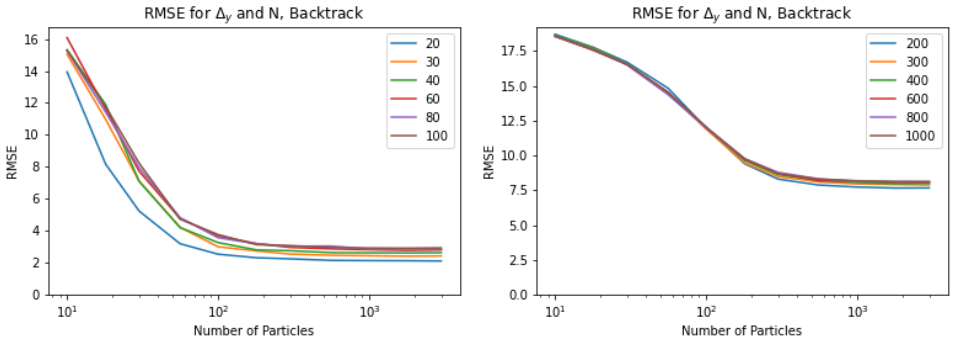

Figure 2.5: Particle Backtracking $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots made with the Particle Backtracking Method give the value of RMSE versus the number of particles in the simulation. Observations on x and observations on y are used in this filter. Each line color is used for a different y proxy timescale. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right.

This method has a similar tendency to gain precision with the number of particles like the Base Method. It is more accurate for lower than for higher Δ_y values when $\Delta_x = 10dt$. The value of Δ_y has however a very negligible effect on the performance of the filter on higher timescales (when $\Delta_x = 100dt$).

2.3.3 Cumulative Resampling

The Cumulative Resampling Method resamples at low frequency only. It keeps the likelihoods for high frequency and multiplies them with the low frequency ones before resampling.

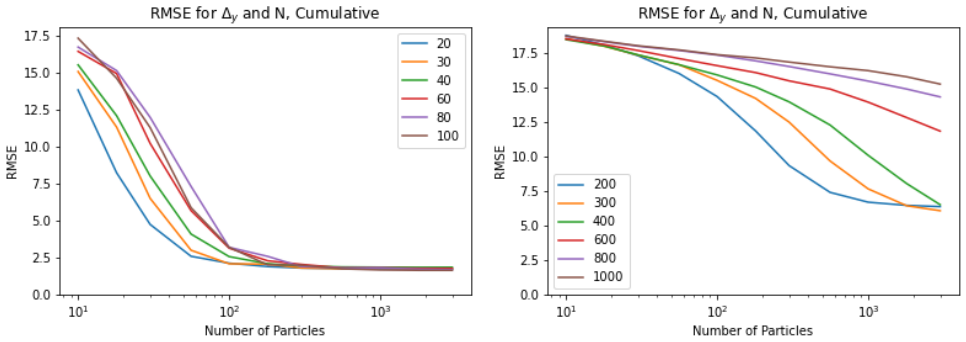

Figure 2.6: Cumulative Resampling Method $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots made with the Cumulative Resampling Method give the value of RMSE versus the number of particles in the simulation. Observations on x and observations on y are used in this filter. Each line color is used for a different y proxy timescale. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right.

As can be seen on the plots of Figure 2.6, the quality of reconstruction, at a low number of particles, is best at low values of Δ_y. However, past a certain threshold of number of particles, the higher frequency reconstructions start to show better reconstruction quality than lower frequency ones. This can be clearly seen in Figure 2.6, on the right plot, where the yellow curve ($\Delta_y = 300dt$) goes under the blue one ($\Delta_y = 200dt$) when the number of particles is beyond 10^3.

2.4 Discussion on the First Three Methods

2.4.1 Base Method: Compared to RMSE of the Free Reconstruction

In order to construct a proper framework for our analysis, the first step is to assess the performance of the Base Method. Doing so requires us to compare its reconstruction with the "free" reconstruction trajectory, where the particle trajectories are unaltered and no resampling is done throughout the entire simulation. We can draw the following comparison:

Method	**10**	**30**	**300**	**3000**
Base	15.65	8.01	3.14	2.97
Free	15.07	14.66	14.41	14.39

Method	**30**	**300**	**3000**
Base	16.10	8.70	8.11
Free	14.63	14.44	14.41

Table 2.1: RMSE tables for the Base Method and the Free Reconstruction N, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two tables give the general picture as to how the performance of the Base Method compares to the Free Reconstruction. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right.

In Table 2.1, the comparisons between the Base Method RMSE's and the Free Reconstruction RMSE's show that, with the exception of very low particle numbers (10 particles for $\Delta_x = 10dt$ and 30 particles and under for $\Delta_x = 100dt$), the Base Method is performing better than the Free Reconstruction. Thus, the Base Method is a better benchmark than the Free Reconstruction. Hence, we will use it to evaluate the performance of the Particle Backtracking and the Cumulative Resampling methods.

2.4.2 Particle Backtracking: Compared to RMSE of the Base Method

Δ_y \ N	30	300	3000
20	5.23	2.24	2.10
40	7.09	2.75	2.62
100	8.23	3.07	2.93

Δ_y \ N	30	300	3000
200	16.69	8.29	7.66
400	16.57	8.62	7.99
1000	16.54	8.68	8.11

Table 2.2: RMSE tables for the Particle Backtracking Method and N, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

The two tables give the RMSE values obtained with the Particle Backtracking Method based on Δ_y and the number of particles in the simulation. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right. The background colour of a cell gives information about how the performance of the Particle Backtracking Method compares to the performance of the Base Method, whether it is better (green if the RMSE is at least 10% smaller), equal (white if the RMSE is between -10% and +10%), or worse (red if the RMSE is at least +10%).

This filter works very well with small values of $\frac{\Delta_y}{\Delta_x}$. However, its performance improvement is less significant as compared to the Base Method when the ratio between Δ_y and Δ_x is higher. This is likely due to two reasons:

1. The bigger the gap between the two assimilation frequencies, the more we get a restricted genealogy in between the particles and reduce the pool of parent particles. We therefore get a worse representation of the particle swarm, contributing to a less efficient reconstruction.
2. Following our noise scheme, more information is contained in a series of multiple observations at high frequency than one observation at low frequency covering the same time lapse. This might explain the quality of reconstructions at high frequency Δ_y versus low frequency.

The Particle Backtracking method's performance is quite good. In fact, it is never lower than the Base Method and is quite robust against degeneracy. This is promising regarding its scalability on EMICs.

2.4.3 Cumulative Resampling: Compared to RMSE of the Base Method

Δ_y \ N	30	300	3000
20	4.75	1.80	1.75
40	8.03	1.95	1.85
100	11.31	2.00	1.65

Δ_y \ N	30	300	3000
200	17.27	9.32	6.36
400	17.33	13.93	6.50
1000	17.99	16.81	15.22

Table 2.3: RMSE tables for the Cumulative Resampling Method and N, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two tables were made with the Cumulative resampling method, they give the value of RMSE based on Δ_y and the number of particles in the simulation. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right. The background colour of a cell gives information about how the performance of the Cumulative Resampling Method compares to the performance of the Base Method, whether it is better (green if the RMSE is at least 10% smaller), equal (white if the RMSE is between -10% and +10%), or worse (red if the RMSE is at least +10%).

The Cumulative Resampling Method proves to be very efficient with $\Delta_x = 10dt$ as a whole. It is especially good at reconstructing for a high value of Δ_y and a high number of particles while also being especially bad with a low number of particles and a high value of Δ_y. This

highlights the fact that this filter acts as a double edged sword. On one hand, it is very efficient at reconstructing big portions of the simulation when the sample size is very big because some particles will have evolved very closely to the true trajectory for a long time. On the other hand, it is very prone to degeneracy when the sample size is too small. Indeed, this is confirmed by the table for $\Delta_x = 100dt$. In that specific case, the degeneracy risk is greater as a whole because observations are further apart. This explains why this filter struggles for most cases, unless it has both a short resampling time and a large number of particles.

This makes the Cumulative Resampling Method the best method asymptotically. It will however be challenging to scale it to EMICs. In such models, degeneracy is very common and a filter that needs to stay on track to function properly is likely going to struggle.

To sum up, both the Particle Backtracking and the Cumulative Resampling methods have advantages and disadvantages when it comes to reconstructing the trajectory as compared to the Base Method. The Particle Backtracking Method is always at least as good as the base method, but is never as good as the Cumulative Resampling Method when the latter is better than the Base Method. The Cumulative Resampling Method excels when it has enough particles but is inclined to degeneracy. All this taken into consideration, it is tempting to seek a way to combine the Cumulative Resampling and Particle Backtracking methods in order to synergize the advantages of both. This will be discussed in the following section on Conditional Resampling.

2.5 Exploration and Discussion of Conditional Resampling

Conditional Resampling can be perceived as a candidate to get the best of both worlds with Cumulative Resampling and Particle Backtracking. The trick comes with devising a way to get this form of resampling to work as either method when it is preferable for the reconstruction. We need to find a condition, on the likelihoods, from which we can decide whether or not to resample early or to finish the Δ_y chunk. In order to do so, it is important to devise the threshold for which Cumulative Resampling (which is prone to divergence) should be switched for the Particle Backtracking approach.

When thinking of a numerical way to qualify the distribution of the likelihoods in the likelihood vector, entropy quickly arises as a promising candidate. Entropy is a measure of the dispersal of a certain quantity in a certain configuration. Here, it can be applied to calculate the average level of uncertainty in the likelihood vector. In information theory, entropy of a discrete probability distribution X is calculated using the following:

$$\mathrm{H}(X) = -\sum_{i=1}^{N} \mathbf{P}(x_i) \log \mathbf{P}(x_i) \tag{2.10}$$

The logarithm in this formulation is by default the natural logarithm, but it can virtually be of any base. In our case, having a logarithm in the N base can be convenient since it means the entropy of the N-length vector oscillates between 1 (case where all particles have likelihoods equal to $\frac{1}{N}$) and 0 (when all the weight is on one particle).

In order to determine whether or not a certain cutoff entropy value could be used in our case, one should evaluate the average entropy of the likelihood vector when resampling in either method. This gives the following data for each configuration:

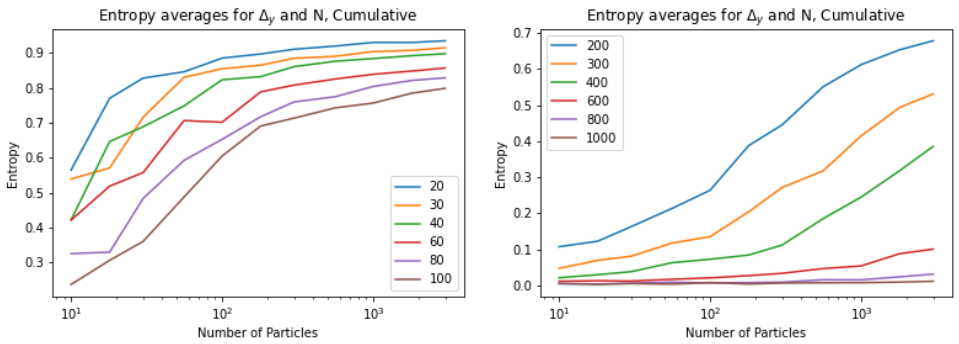

Figure 2.7: Average entropy of the likelihood vector at resampling with the Cumulative Resampling method, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots made with the Cumulative Resampling Method give the average value at resampling of the entropy of the likelihood vector. Observations on x and observations on y are used in this filter. Each line color is used for a different y proxy timescale. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right. This figure was averaged over only 10 runs.

As can be seen in Figure 2.7, the entropy of the likelihood vector seems to evolve in a predictable pattern with the setup of the experiment. Indeed, it seems to be possible to determine a threshold at which to switch from one method to the other. This would allow us to have it behave in a certain manner, taking on the preferable traits when useful.

The following figures, done with 100 runs, show the evolution of the RMSE when we put a threshold on the entropy of the likelihood vector as a resampling condition. We call this threshold α, value under which resampling is required.

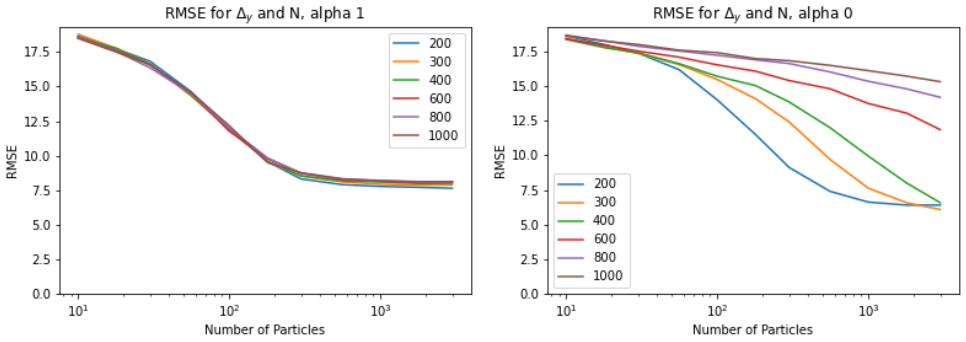

Figure 2.8: Average RMSE in Conditional Resampling for different entropy thresholds for $\Delta_x = 100dt$, $\alpha = 1$ (left) and $\alpha = 0$ (right)

These two plots illustrate the performance of the Conditional resampling method using different entropy criteria thresholds. They were both performed on the $\Delta_x = 100dt$ timescale. Each line color is used for a different y proxy timescale. The threshold on entropy of the likelihood vector is 1 on the left and 0 on the right.

The entropy criterion proves to be effective since when $\alpha = 1$ (as shown on the left in Figure 2.8), Conditional Resampling behaves like the Particle Backtracking Method (as shown on the right in Figure 2.5). While when $\alpha = 0$ (as shown on the right in Figure 2.8), the assimilation behaves in a Cumulative Resampling fashion (as shown on the right in Figure 2.6). Finding a criterion value which can get us the best of both worlds can be done empirically. A value of $\alpha = 0.85$ offered results for a promising analysis.

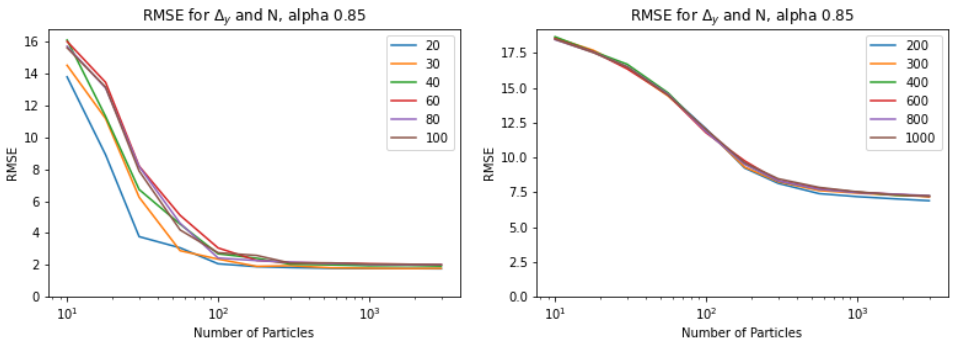

Figure 2.9: Average RMSE for Conditional Resampling based on entropy with $\alpha = 0.85$, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two plots display the performance of the Conditional Resampling method using a threshold of 0.85. Each line color is used for a different y proxy timescale. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right. These two plots were averaged over only 10 runs.

Δ_y \ N	30	300	3000
20	3.77	1.83	1.76
40	6.73	2.04	1.92
100	7.89	2.11	2.01

Δ_y \ N	30	300	3000
200	16.47	8.14	6.89
400	16.69	8.30	7.21
1000	16.50	8.47	7.23

Table 2.4: RMSE tables for Conditional Resampling based on entropy with $\alpha = 0.85$, $\Delta_x = 10dt$ (left) and $\Delta_x = 100dt$ (right)

These two tables were made with the Conditional Resampling method based on entropy with $\alpha = 0.85$. They give the value of RMSE based on Δ_y and the number of particles in the simulation. High frequency observations (on x) are performed every 10 timesteps on the left and every 100 timesteps on the right. The background colour of a cell gives information about how the performance of the Conditional Resampling Method compares to the performance of the Base Method, whether it is better (green if the RMSE is at least 10% smaller), equal (white if the RMSE is between -10% and +10%), or worse (red if the RMSE is at least +10%).

Figure 2.9 and Table 2.4 show the performance of assimilation using Conditional Resampling based on entropy when we take $\alpha = 0.85$. The effect of this new criterion might seem small, but it is significant. Indeed, in Figure 2.9 the shape of the curves show that the method clearly first behaves as Particle Backtracking and switches to Cumulative Cumulative as the number of particles increases. This effect can be further observed when looking into the performance tables of the method (Figure 2.9). Indeed, when compared to its parent tables (Table 2.2 and 2.3) we observe that not only does the Conditional Resampling method inherit from the strengths of both methods, but it also does better than both of them in several setups. This can be seen (in Table 2.4 on the right), where both parent methods were not significantly better than the Base Method in the setup ($\Delta_y = 1000$, $N = 3000$) while the Conditional Resampling Method is.

2.6 Correlation Results

The RMSE gives an idea about how close or how far the reconstructed trajectory was to the reference trajectory. However it lacks insight as to what parts of the trajectory were properly reconstructed and what parts were left out. In our case, it can be really useful to determine how well each reconstructed variable changes with respect to its counterpart in the reference simulation. Thus depicting how reconstruction evolves across variables from one set of parameters to another for a specific method. In order to better illustrate the

tendencies observed we will not cover all the setups in the multivariate analysis. We will arbitrarily choose setups where:

- $\Delta_x = 100dt$
- $\Delta_y = 2\Delta_x = 200dt$
- $N = 10, 100, 1000$

We will also measure the correlation coefficients over different time series (following Equation 2.9):

- Time series of trajectories over every timestep
- Time series of trajectories on averages over Δ_x timesteps (the smallest scale of assimilation)

Analysing reconstructions over time averages is often what is done in heavily chaotic systems like climate models, hence the decision to explore this procedure on a smaller scale system.

2.6.1 Proxy Correlations on Δ_x

Since we have proxy values given on Δ_x, we can measure how much they correlate with the actual trajectory averages. As a reminder, our gaussian noise is of $\sigma_n = 2\sqrt{10}$, hence reduced to $\frac{2}{\sqrt{10}}$ for a proxy of $100dt$. For this gaussian noise, we get a correlation value averaging around 0.995 between proxy on x and reference trajectory averages.

2.6.2 Free Reconstruction

The Free Reconstruction made without assimilation shows values of correlation between the reconstruction and the reference curve close to 0 for all variables. If we are to look at it in more detail, the correlation values range around 10^{-5} and tend to be positive. This very small yet positive correlation is likely due to the fact that particles originate from around the starting point in the reference run.

2.6.3 Base Method

For the Base Method with and without interpolation, the correlations of the different time series are given in Tables 2.5 and 2.6.

ρ \ N	**10**	**100**	**1000**
x	0.258	0.748	0.896
y	0.219	0.685	0.85
z	0.066	0.419	0.677

ρ \ N	**10**	**100**	**1000**
x	0.443	0.925	0.996
y	0.44	0.918	0.992
z	0.104	0.523	0.809

Table 2.5: Correlation tables for the Base Method without interpolation, time series of step dt (left) and Δ_x (right)

These two tables were made with the Base Method without interpolation. They give the value of correlation based on the variable in question and the number of particles. The correlation is calculated on time series for one value every dt on the right and on averages over Δ_x on the left.

ρ \ N	10	100	1000
x	0.151	0.285	0.408
y	0.127	0.235	0.335
z	0.023	0.045	0.07

ρ \ N	10	100	1000
x	0.295	0.576	0.782
y	0.289	0.558	0.752
z	0.04	0.092	0.145

Table 2.6: Correlation tables for the Base Method with interpolation, time series dt (left) and Δ_x (right)

These two tables were made with the Base Method with interpolation. They give the value of correlation based on the variable in question and the number of particles. The correlation is calculated on time series for one value every dt on the right and on averages over Δ_x on the left.

The first takeaway from Tables 2.5 and 2.6, is that the correlations show some degree of assimilation in all setups explored. While the Free Reconstruction would always show no correlation, divergent filtering methods will still show some degree of it.

We also have for both methods a tendency to have a better reconstruction on x and y than on z. This is especially true for the Base Method with interpolation. Since it was seen previously that the particle swarm in the Base Method with interpolation was very often divergent, it can be expected to have very poor performance on variables which are not observed. As for the Base Method without interpolation (Table 2.5), the lack of correlation on the z variable (sometimes around 0.8 on z vs around 1 on x and y) shows that the information given by the proxies is not enough for the system to follow all components of the reference trajectory.

The correlation analysis still helps us in uncovering a striking occurrence: the correlations shown by the Base Method with interpolation on x and y is quite high. This can come as a surprise considering the fact that they did not show any degree of assimilation with $\Delta_x = 100dt$ in the RMSE measurements. This shows that regardless of how lost in a state space a particle swarm can be, reconstruction attempts from observation variables will still have an effect on the correlation between reference and reconstruction.

An overarching tendency is that correlation is systematically better over time series on Δ_x than for every timestep.

2.6.4 Particle Backtracking

Particle Backtracking was shown to behave very similarly to the Base Method in chaotic setups when looking at the RMSE. This tendency is also found in the correlations as can be seen in Table 2.7 which shows little to no improvement compared to Table 2.5 despite having an additional assimilation on y proxies.

ρ \ N	10	100	1000
x	0.241	0.727	0.913
y	0.204	0.667	0.871
z	0.068	0.406	0.711

ρ \ N	10	100	1000
x	0.424	0.904	0.997
y	0.426	0.899	0.993
z	0.097	0.507	0.839

Table 2.7: Correlation tables for the Particle Backtracking Method, time series dt (left) and Δ_x (right)

These two tables were made with the Particle Backtracking Method. They give the value of correlation based on the variable in question and the number of particles. The correlation is calculated on time series for one value every dt on the right and on averages over Δ_x on the left.

2.6.5 Cumulative Resampling

The Cumulative Resampling Method was shown to struggle a little when faced with more degrees of freedom between resampling steps. This is still observable in Table 2.8 where it

struggles in reconstructing all of the system variables correctly with z correlations lagging behind the x and y correlations. As expected, with enough particles this is less noticeable.

ρ \ N	10	100	1000
x	0.244	0.594	0.935
y	0.208	0.538	0.903
z	0.059	0.31	0.8

ρ \ N	10	100	1000
x	0.427	0.799	0.996
y	0.429	0.796	0.994
z	0.081	0.356	0.871

Table 2.8: Correlation tables for the Cumulative Resampling Method, time series dt (left) and Δ_x (right)

These two tables were made with the Cumulative Resampling Method. They give the value of correlation based on the variable in question and the number of particles. The correlation is calculated on time series for one value every dt on the right and on averages over Δ_x on the left. Similarly to the three previous methods, correlation is systematically better over time series on Δ_x than on every timestep.

2.7 Overall Comments on Correlation Observations

The analysis of the correlations on the time series allowed us to gain some precious insights to keep in mind for the analysis with the application on LOVECLIM. Namely:

- A divergent simulation can still properly reconstruct some variables, especially those which are observed.
- Time series scaling matters, with time series over long steps gaining in correlation by comparison with finer steps.

2.8 Complementary Visual Exploration

It is often useful to visualise the behaviours of particles by plotting their trajectories throughout a portion of the simulation. This highlights how the observation influences the particles and, as a result, the final reconstruction. This section shows how the particle swarms behaves when it is subject to different methods. It helps in explaining the results we see above.

We will focus on three methods only, which are the most relevant to our large scale experiment. These are the Base Method without interpolation, the Cumulative Resampling Method and the Particle Backtracking Method. Furthermore, it is important to note that this section illustrates the behaviour of each method based on only one of its runs for visualisation purposes. Taking only one run helps considerably in simplifying the analysis. However, it is bound to be a little biased since all runs are different because their proxy noise and the randomness of the particles are non-deterministic.

We will show the behaviour of the assimilation from timesteps 0 through 600, with base parameters $N = 100$ and $\Delta_y = 3\Delta_x$.

The figures will feature:

- In black scatter: the proxy average value of that variable given over Δ_d
- In blue: the true trajectory of the system
- In red: the fertile particles (particles with daughters) trajectories
- In grey: the infertile particles (particles without daughters) trajectories
- In green: the weighted average of particles, the reconstructed trajectory

Base Method without interpolation

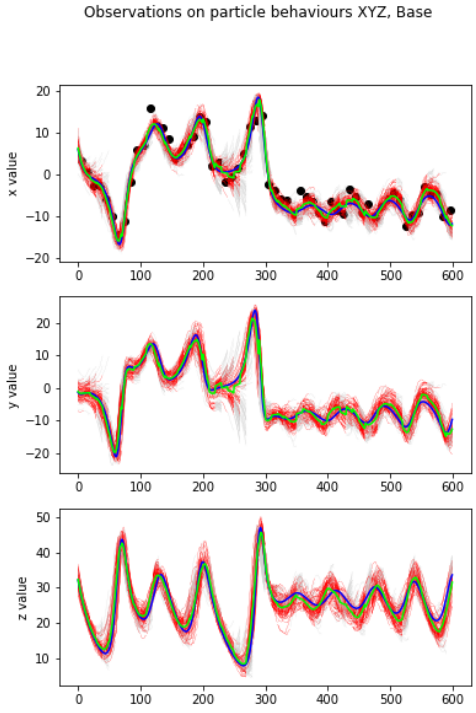

Figure 2.10: XYZ plots, $\Delta_x = 10$

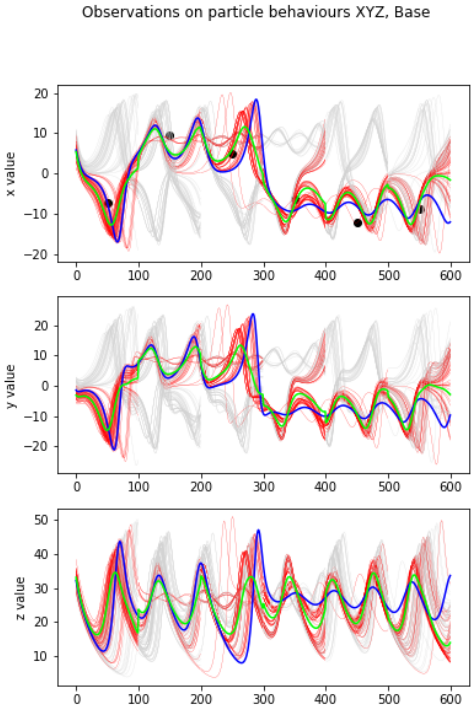

Figure 2.11: XYZ plots, $\Delta_x = 100$

As can be observed in these plots, the Base Method works much better at reconstructing the trajectory at $\Delta_x = 10dt$ than at $\Delta_x = 100dt$. The difference seems to be clear cut. At high frequency, the reconstruction with 100 particles is very close to the actual value of the function, being sometimes diverted by extreme observations. It is important to note that the true trajectory of particles in the $\Delta_x = 100dt$ case is not that far from the actual tendency of the curve.

Cumulative Resampling

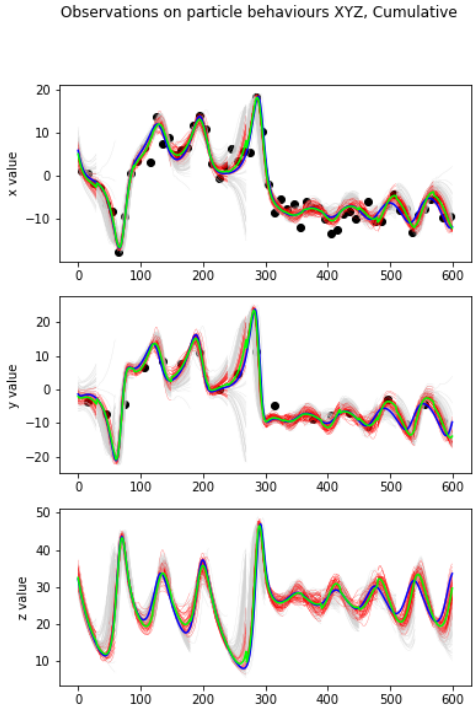

Figure 2.12: XYZ plots, $\Delta_x = 10$

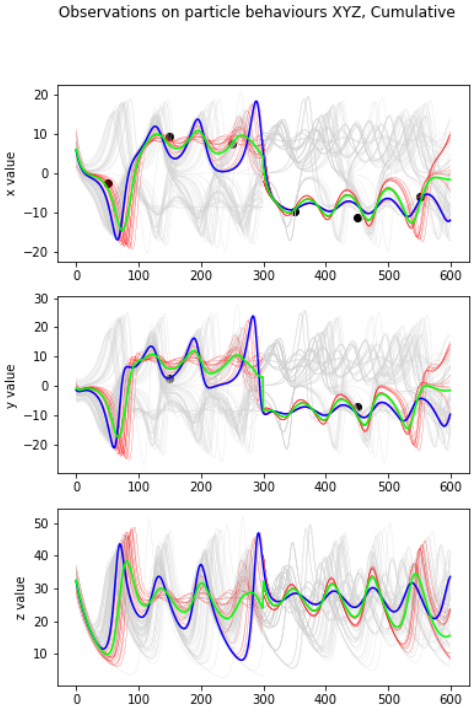

Figure 2.13: XYZ plots, $\Delta_x = 100$

In these plots, it can be seen that Cumulative Resampling is slightly less subject to nudging from noisy observations than the base method (Figures 2.12 and 2.10). In Figure 2.13 where $\Delta_x = 100dt$, there are a lot of infertile particles, showing how most particles evolve into inaccurate directions when there are more degrees of freedom between resamplings, which further highlights the filter's tendency to degenerate.

Particle Backtracking

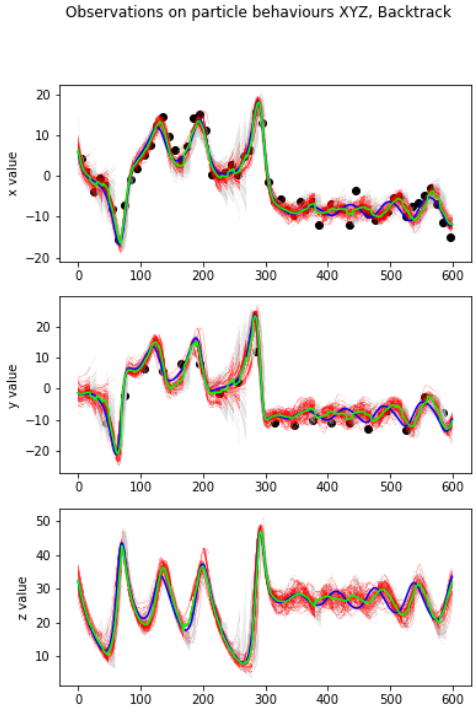

Figure 2.14: XYZ plots, $\Delta_x = 10$

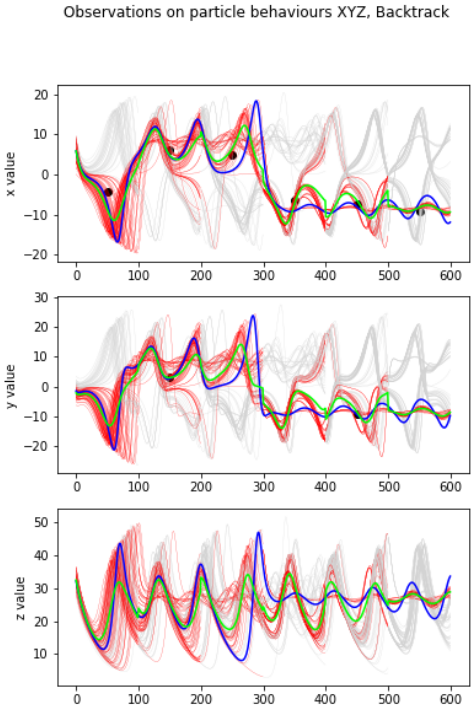

Figure 2.15: XYZ plots, $\Delta_x = 100$

On these plots, it is quite explicit that the Particle Backtracking Method behaviour resembles the Base Method behaviour. It still seems as though it has more infertile particles at resampling than the Base Method. This could be due to the stricter resampling which takes into account two likelihoods at low frequency.

Chapter 3

Applications to LOVECLIM

In order to establish whether or not the tested methods can be used in paleoclimatology, we will test them on an intermediate complexity model, LOVECLIM. Using LOVECLIM will allow to draw additional diagnostics as to the performance of the different methods on a more practical model. This document will present the results for the Base Method (with and without interpolation) and the Cumulative Resampling Method. The results for the Particle Backtracking Method and the Conditional Resampling Method are not yet available.

3.1 The LOVECLIM Model

All information about LOVECLIM in this section is available in paper [3]. LOVECLIM is a three dimensional EMIC which has been cited in hundreds of papers as of today. Its current version couples the dynamics of five components at most:

- **Atmosphere** - `ECBilt`: This component models the atmosphere with a grid resolution of about 5.6° in latitude and longitude (32×64), as well as having three vertical truncation levels.
- **Sea ice and ocean** - `CLIO`: This component was the first to be coupled with the atmosphere component. It is of a 3° resolution both in latitude and longitude (65×120), being in itself the coupling between a comprehensive sea ice model and an ocean GCM.
- **Continental biosphere** - `VECODE`: The terrestrial biosphere can have a strong impact on the dynamics of the climate, having effects on air composition, surface albedo, humidity and many others. Vegetation is assumed to cover a proportion of given areas, being composed of either grass or forest.
- **Ocean carbon cycle** - `LOCH`: The ocean carbon cycle is thought to be a major driving agent in atmosphere CO_2 concentration. Since CO_2 acts as a greenhouse gas, drawing its dynamic precisely is important if we want to model the climate.
- **Polar ice sheet** - `AGISM`: This component simulates the thermomechanical ice dynamic for Antarctica and Greenland.

The components `LOCH` and `AGISM` will not be used in our runs.

LOVECLIM model takes climate forcings into account as well. Forcings are external drivers of the climate, independent factors which have an effect on the climate system as a whole. Examples of forcings include [37]:

- Solar radiation output (natural)
- Volcanic eruptions (natural)
- Changes in atmospheric concentration of greenhouse gases (natural/anthropogenic)
- Aerosols (natural/anthropogenic)

3.2 Experimental Setup

In order to observe the behaviour of the different methods on a model far from the simplicity of Lorenz63, it is important to establish a rigorous experimental procedure. In doing so, we will be able to draw relevant conclusions as to the quality of reconstruction of the different methods.

3.2.1 Methodology

The experimental methodology in this section is similar to the one that was used for the tests on Lorenz63.

1. We run a deterministic LOVECLIM reference run
2. We draw noisy pseudoproxies from different variables in that run
3. We run particle filtering methods on LOVECLIM assimilating for the pseudoproxies
4. We try to reconstruct the reference run by taking a weighted mean of the particles based on their resampling coefficients after every assimilation

3.2.2 Choosing the Climate Setting

We wanted to minimise the effects forcings can have on the system because it may tamper with the performance assessment. Two separate climate simulations run over the same time period on the same model will likely be uncorrelated. However, the more forcings are included, the more these two climate runs will be "forced" into similar states. We would therefore observe correlation between different runs. Our particles are run on the same parameters as our reference simulation. We want to minimise their correlation on the accounts of external factors as much as possible by limiting the effects induced by forcings.

Industrial climate reconstructions need to take into account the forcing induced by greenhouse gases and aerosols concentration in the atmosphere, which can have a very heavy effect on the dynamics of the climate. Therefore, we decided to opt for a pre-industrial climate reconstruction. The elected period spans between 1000 and 1200 A.D (around 8 hours of CPU time on one single Xeon processor (2.5 Ghz)[3]).

3.2.3 Creating the Pseudoproxies

Electing the location

The first step in testing the methods is to establish the observation variables and their respective noise. Climate models have many different variables over very large grids. We want to establish the location of observation as well as the variable of observation. In this experiment, we decided to take pseudoproxies on decadal Sea Surface Temperatures (SST) and yearly atmospheric Surface Temperatures (TS). The locations at which these pseudoproxies were made are (in geographical coordinates):

°N	°E
58.21	-29.57
67.45	-17.09
65.29	4.84
62.69	-67.69
70.52	-54.13
78.59	2.36

Table 3.1: SST pseudoproxy locations

°N	°E
41.53	0
47.07	11.25
69.21	22.5

Table 3.2: TS pseudoproxy locations

These locations are all in the upper northern hemisphere and concentrated around the eastern Atlantic. The map in Figure 3.1 shows these pseudoproxy locations. These locations

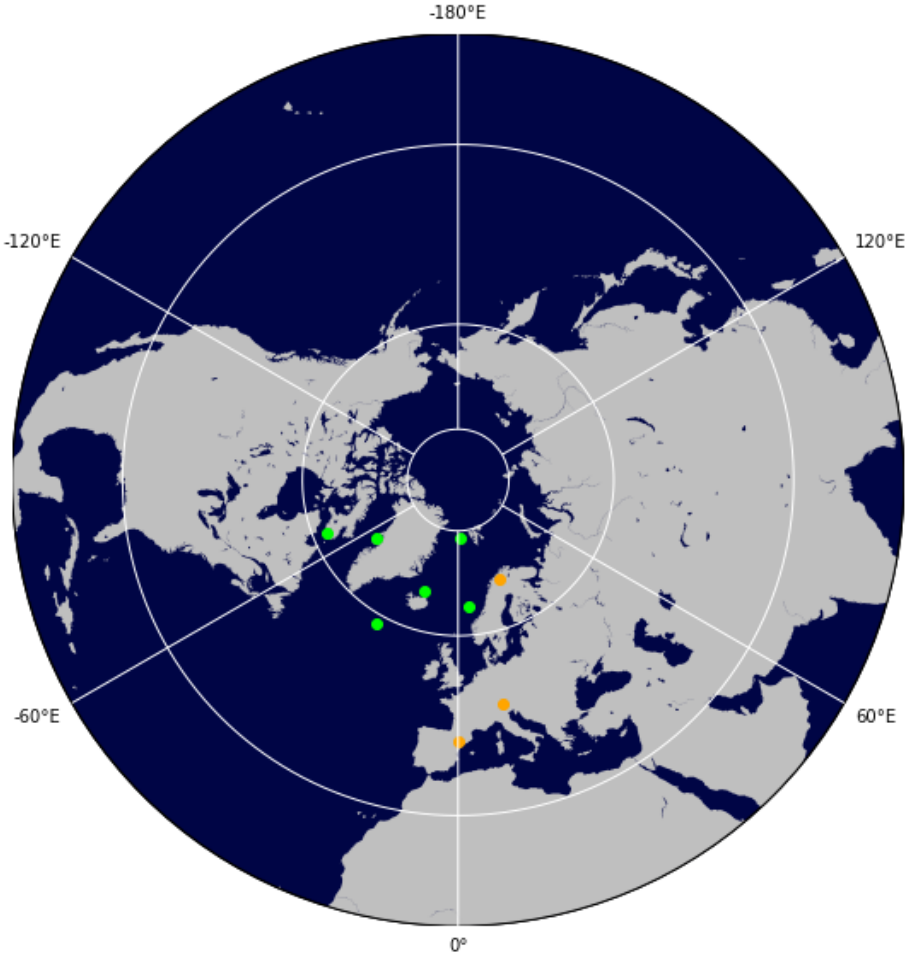

Figure 3.1: Illustration of the pseudoproxies, their locations and types

This is a plot of the northern hemisphere in a North stereographic projection with the chosen locations for each pseudoproxy. The locations are shown in green for SST pseudoproxies and in orange for TS pseudoproxies.

were chosen because they are in the vicinity of authentic proxy sampling sites. At the terrestrial proxy locations one can find tree ring samples [29, 30, 31]. At the ocean proxy sites, we get biological sediment proxies like alkenones and dinocyst [32, 33, 34, 35, 36]. For the sake of our assimilation, their geographical location was reduced to the centres of cells in the LOVECLIM grid model, giving the locations for the pseudoproxies.

For the sake of simplicity, as we will only focus on pseudoproxies from now on, we will refer to them as proxies throughout this chapter of the document.

Determining the noise

The choice for the noise on these pseudoproxies was done based on the natural variability of each pseudoproxy target variable. The making of the pseudoproxy is done by going through the following steps:

1. Averaging: We reduce the table of monthly values of a certain proxy into a table of $\Delta_v dt$ time averages. This effectively gives an array containing time averages at this proxy's time resolution.
2. Calculating natural variation: We get the natural variation of proxies in a certain distribution by taking the standard deviation of the proxy table found in step 1. This standard deviation gives a measure of the quantity by which a proxy state variable fluctuates.

3. Computing the noise: Now that we have the value for the standard deviation, it gives a proper scalability factor for the noise. We calculate the noise to the proxy sample by making a vector of samples from a normal distribution (same length as our time average vector from step 1) with a standard deviation equal to the standard deviation found in step 2 and an average of 0.
4. Adding the noise: The final step into making a noisy pseudoproxy is by adding the noise which was computed in step 3 to the time averages computed in step 1.

3.2.4 Methods tested

In order to have a reference for performance, we have simulated 10 free runs on the period of interest with no assimilation. This provides an average performance from several free "reconstructions" which will enable evaluation of the performance of the tested methods.

The tested methods on the LOVECLIM model are:

- A Base Method assimilating the high frequency proxies (here the land proxies). This method does not assimilate the ocean proxies at all.
- A Base Method interpolating the low frequency proxies to the high frequency. Here the interpolation is a step interpolation (e.g, proxy says average 5°C in a location for 10 years, interpolation says 5°C average every year on that period)
- The Cumulative Resampling Method observing the proxies at their respective timescales and assimilating them at a resolution of 10 years (low frequency).

All methods will be tested with 96 particles.

3.2.5 Evaluating the Reconstruction

In this section, we will mostly use correlation coefficients between the reconstruction and the reference simulation in different grid areas to determine the quality of the reconstruction. The RMSE can be computed on temperatures as well, however it proved to be quite inefficient in this case, especially for the visualisation (See Appendix .3). Hence, in our case, we will use the correlation coefficient to describe how well the reconstruction approximates the reference run.

The correlation will be evaluated and averaged over different parts of the LOVECLIM grids (geographical regions) containing the proxy locations:

- We calculated the punctual correlations at proxy locations between the reference run and the reconstructions and averaged the correlations. This gives an idea on the quality of reconstruction in discrete geographical locations with observations (proxy locations). The following analyses were performed:
  - On SST, in `CLIO` grid (the sea ice and ocean component)
    - At SST proxy locations (Figure 3.2)
  - On TS, in `ECBilt` grid (the atmosphere component)
    - At TS proxy locations (blue in Figure 3.3)
    - At SST proxy locations (red in Figure 3.3)
    - At TS and SST proxy locations
- We calculated the punctual correlations in zones around proxy locations between the reference run and the reconstructions and averaged the correlations. This gives an idea about the average quality of reconstruction in discrete geographical locations in the vicinity of proxies. The following analyses were performed:
  - On SST, in `CLIO` grid
    - In the zone around SST proxy locations (Figure 3.4)

  - On TS, in `ECBilt` grid
    * In the zone around TS proxy locations (blue in Figure 3.5)
    * In the zone around SST proxy locations (red in Figure 3.5)
    * In the zone around TS and SST proxy locations

- We calculated the correlations between the geographical averages of temperature in zones around the proxies in the reference run and the reconstruction. This gives an idea on the quality of reconstruction on geographical zones in the general vicinity of the proxies. The following analyses were performed:
  - On SST, in `CLIO` grid
    * In the zone around SST proxy locations (Figure 3.4)
    * In the northern hemisphere
  - On TS, in `ECBilt` grid
    * In the zone around TS proxy locations (blue in Figure 3.5)
    * In the zone around SST proxy locations (red in Figure 3.5)
    * In the zone around TS and SST proxy locations
    * In the northern hemisphere

3.2.6 Grid Maps

In this section, we will illustrate the grid maps on which the correlations will be evaluated.

Punctual SST Grid Map

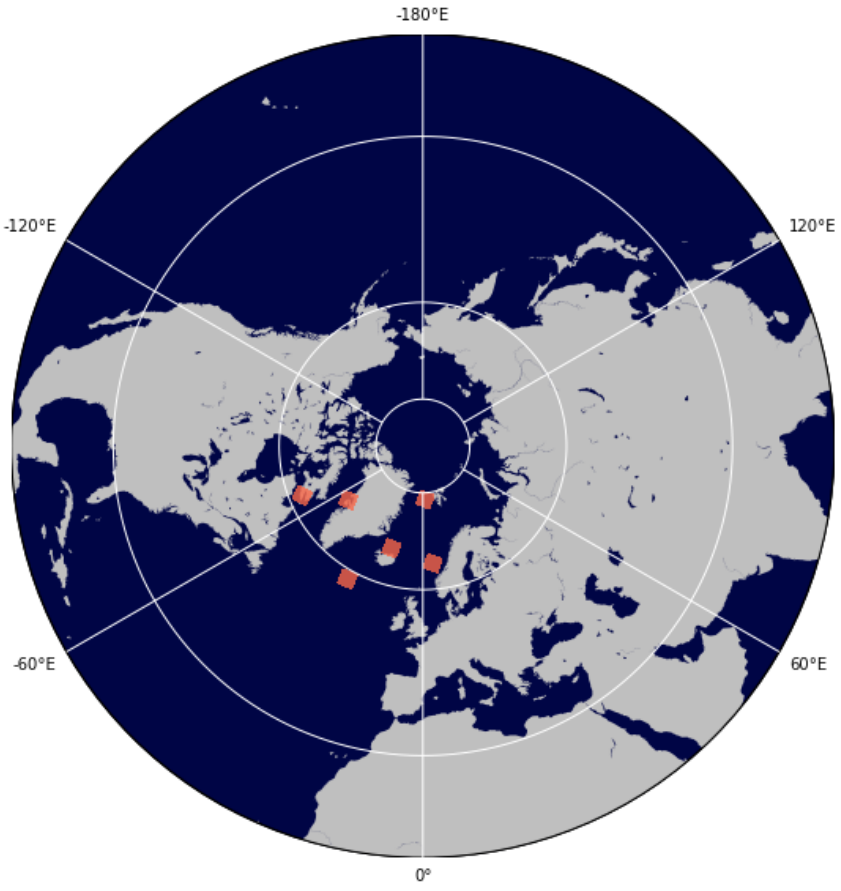

Figure 3.2: Illustration of the proxy areas in SST grid

This plot of the northern hemisphere in a North stereographic projection shows the LOVECLIM ocean grid areas occupied by each respective ocean proxy in red.

In our experimental setup, the SST data were assimilated at 6 points in the northern hemisphere at a resolution of 10 years.

Punctual TS Grid Map

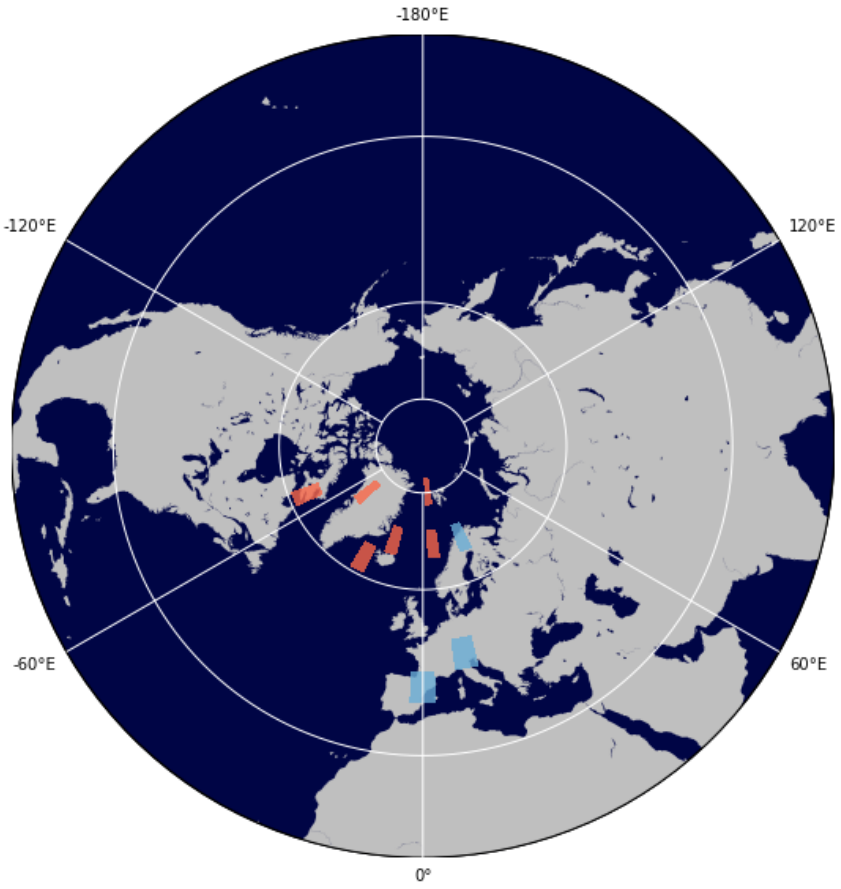

Figure 3.3: Illustration of the proxy areas in TS grid

This plot of the northern hemisphere in a North stereographic projection shows the LOVECLIM atmosphere grid areas occupied by proxies. Grid areas occupied by ocean SST proxies are shown in red while grid areas occupied by land TS proxies are shown in blue.

Atmospheric surface temperatures can be measured on land and on the ocean. Hence, the measurements on TS can be made on land proxies and on ocean proxies. The change to the grid shape of the proxy areas in Figure 3.3 compared to Figure 3.2 comes from the fact that we are now based in the atmospheric grid in LOVECLIM, which is not the same as the oceanic grid.

SST Zone Grid Map

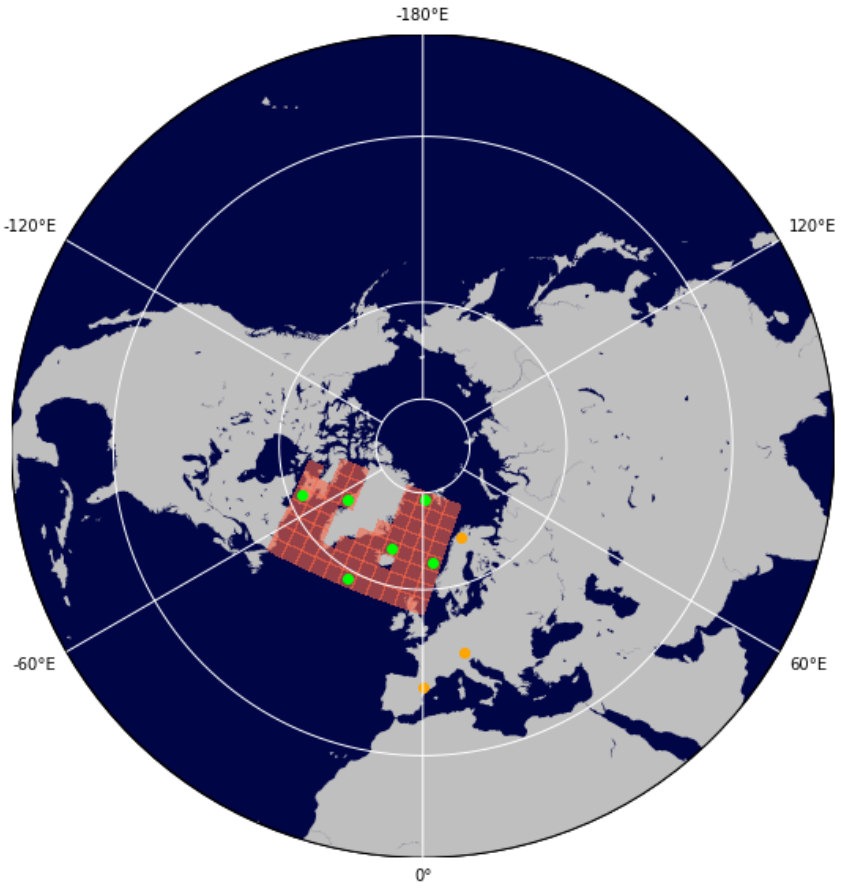

Figure 3.4: Illustration of the oceanic zone around SST proxies

This plot of the northern hemisphere in a North stereographic projection shows the LOVECLIM ocean subgrid containing all ocean proxies.

The zone around SST proxies depicted in Figure 3.4 was taken in this document as the smallest rectangular sub-grid containing all the proxy areas in the reconstruction (it is rectangular with the exception of some terrestrial land areas like Greenland and the Canadian Northern Territories).

TS Zones Grid Map

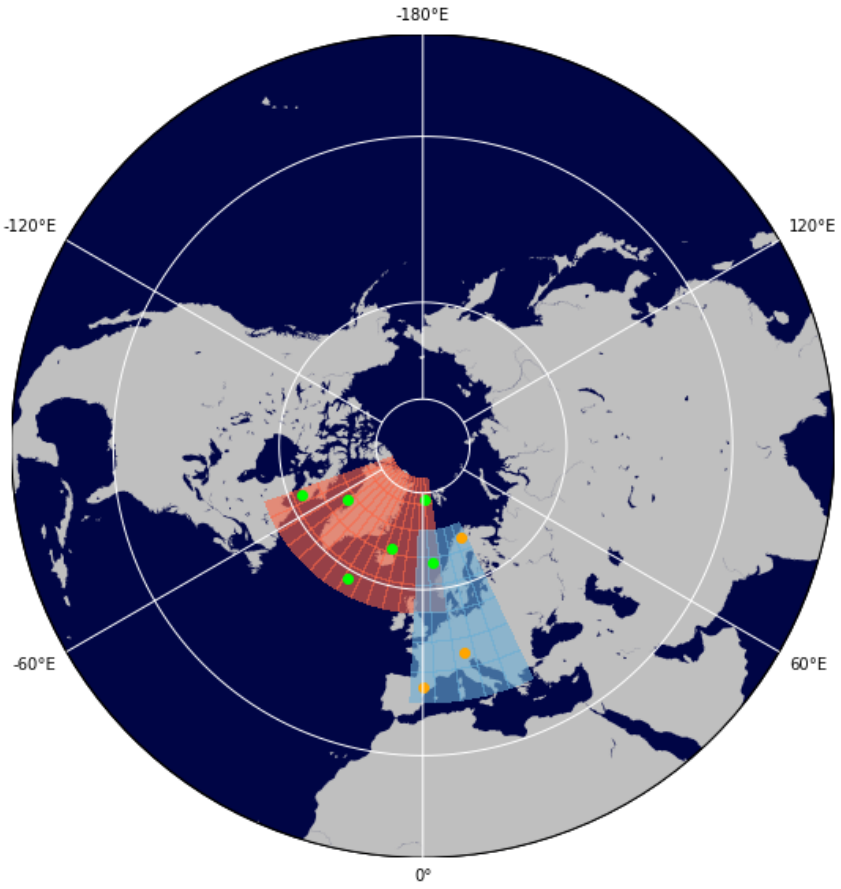

Figure 3.5: Illustration of the atmospheric zones around TS and SST proxies

This plot of the northern hemisphere in a North stereographic projection shows the LOVECLIM atmosphere subgrids containing respectively the ocean SST proxy assimilation zone (in red) and the land proxy TS assimilation (in blue).

3.3 Results

3.3.1 Observation on the Time Series

Prior to delving into a quantitative analysis of the results of each method on LOVECLIM, it can be useful to take a look at a qualitative graph of the performance of the reconstruction by simply visualising the time series.

The time series are average yearly surface temperatures over a zone covering the ocean proxies and the land proxies in the `ECBilt` grid (combination of the red and blue areas in Figure 3.5).

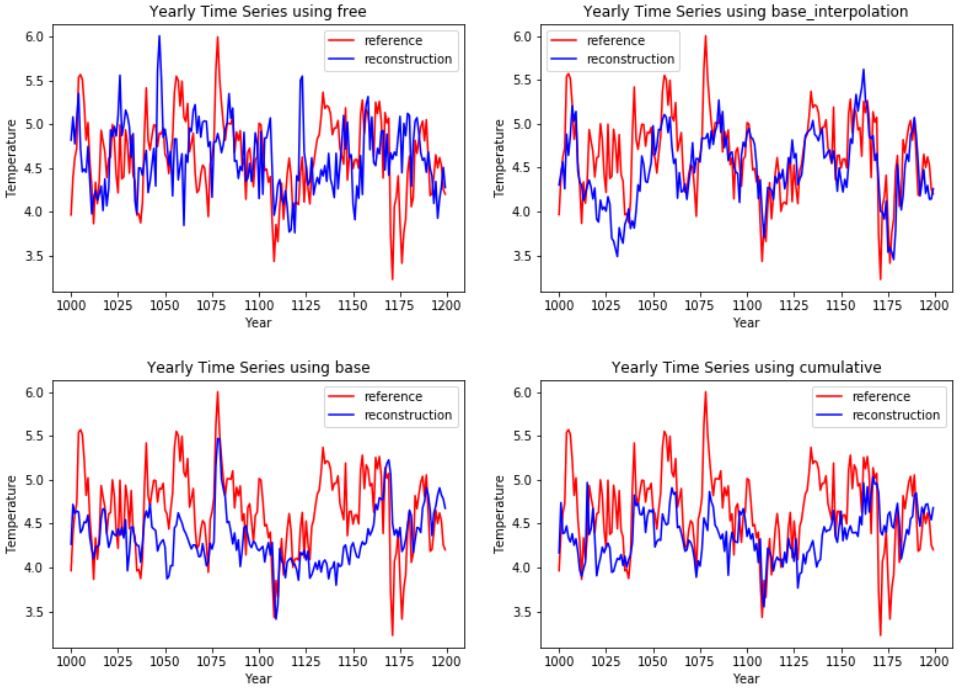

Figure 3.6: Average TS time series of reconstruction on assimilation zone for different methods

These four plots feature the time series of average surface temperature over the zone combining the proxies of interest (combination of the red and blue zones in Figure 3.5). They juxtapose the time series of the reference run (blue) with the time series of reconstructions (red)(upper-left: Free Run, upper-right: Base Method with interpolation, lower-left: Base Method without interpolation, lower-right: Cumulative Resampling Method). The free run is one of the 10 free runs performed on that time period, it was selected arbitrarily.

From the four plots featured in Figure 3.6, we can already make observations on the quality of each reconstruction. The free run is clearly the worst, which is to be expected since there is no assimilation. It seems completely out of sync with the reference trajectory. The other three reconstruction methods all show a certain degree of assimilation, with the Base Method with interpolation looking the best, and the Base Method without interpolation the worst.

3.3.2 Punctual Correlations at Proxies

Calculating the correlations on the proxy grid locations allows to see how well the method reconstructs the system at the points of observation (proxies). Furthermore, at the proxy locations, it is possible to compare the correlations between proxy data and the reference run. This correlation is an additional valuable tool to assess the performance of the reconstruction methods. For instance, if the correlation between a reconstruction and the reference run were higher than the correlation between the reference run and the proxy data, one could conclude that the reconstruction was very successful.

SST punctual correlations at proxies

Figure 3.7 shows that the Base Method with interpolation has the best correlations both at 1 year and 10 years time series. This method is even at least as good as the proxy values at 10 years, with a correlation of 0.82. An encouraging observation is that all methods are better than the free reconstruction, which indicates that the assimilation is making a difference in all the methods used. On this metric, the Base Method without interpolation is at least as good as the Cumulative Resampling Method. This points out that the Cumulative Resampling Method does not seem to benefit from ocean proxy assimilation.

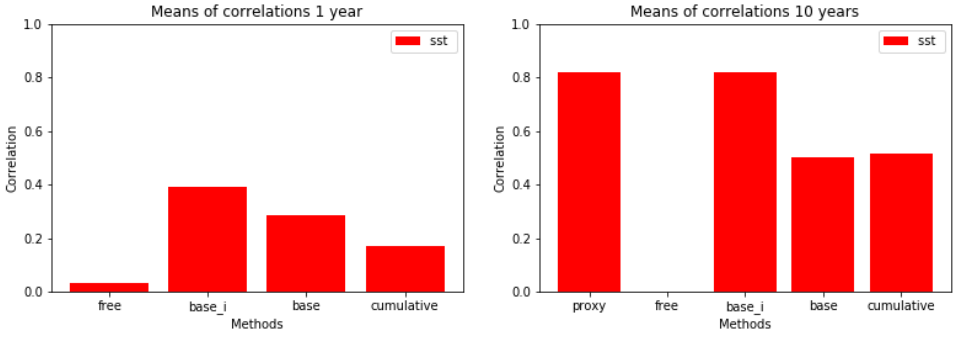

Figure 3.7: SST Correlation Coefficient Averages at SST proxy locations, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the SST at the SST proxy locations as shown in Figure 3.2. The correlations were computed on two different temperature data series, 1 year averages (left), 10 years averages (right).

TS punctual correlations at proxies

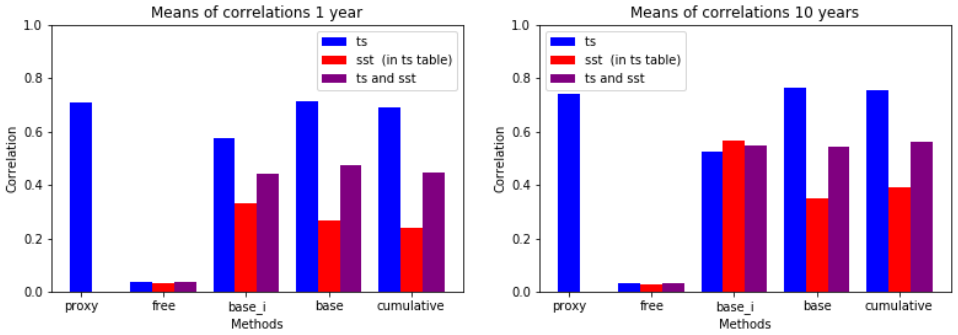

Figure 3.8: TS Correlation Coefficient Averages at SST and TS proxy locations, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the TS at TS proxy locations (blue), at SST proxy locations (red) and at SST & TS proxy locations (purple), as shown in Figure 3.3. The correlations were computed on two different temperature data series, 1 year averages (left), 10 years averages (right).

In Figure 3.8 we can see the average quality of reconstruction based on correlation on TS data at proxy locations. A first general comment is that the free reconstruction shows almost no degree of correlation for any set of proxy locations. Secondly, the assimilation on TS over land proxies has a better correlation coefficient than the SST assimilation over ocean proxies (Figure 3.7) at 1 year. This is true for all methods and can be explained by the higher resolution of TS proxies.

As far as the TS reconstruction over TS proxy locations (blue) is concerned, the performances of the Base Method without interpolation and the Cumulative Resampling Method are equivalent to the quality of the proxies, and this for both timescales. The Base Method with interpolation on the other hand showcases a poorer performance. This can be explained by the fact that this method is prioritising assimilation of ocean proxy data (as it assimilates it 10 times more than the other two methods).

Considering the TS reconstruction over SST proxy locations (red), one can observe a similar pattern than SST reconstructions over SST proxy locations (Figure 3.7). This is because SST and TS are not independent variables. Indeed, the LOVECLIM model dynamics simulate the physical exchanges between atmosphere and ocean. Thus, when a reconstruction focuses

on the SST temperatures, it is bound to show a good reconstruction of the TS at these locations as well.

Finally, the TS reconstruction over TS and SST proxy locations (purple) shows very similar correlation coefficients for all methods and all time series (0.45 for 1 year, and 0.55 for 10 years). The value of this coefficient is systematically in between the average correlation coefficient at TS proxy locations and the average correlation coefficient at SST proxy locations.

3.3.3 Punctual Correlations in Zones around Proxies

We will now focus on zone averages of punctual correlations. This allows to determine the quality of the reconstruction in zones around the proxy locations. In contrast with Section 3.3.2, analysing for zones does not allow comparison with the correlations on the proxies. This is explained by the fact that this analysis involves grid cells with and without proxies. Hence the "proxy" column will not be featured in the bar plots.

SST punctual correlations in zones around proxies

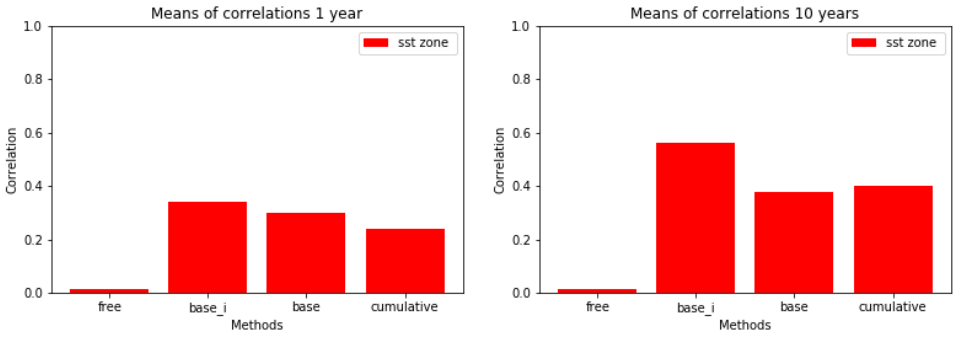

Figure 3.9: SST Correlation Coefficient Averages in the SST ocean zone, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the SST punctual correlations in the SST ocean zone as shown in Figure 3.4. The correlations were computed on two different temperature data series, 1 year averages (left), 10 years averages (right).

The bar plots in Figure 3.9 seem to display very similar characteristics to the ones in Figure 3.7. However, the values of these average correlations are lower with this analysis. This can be expected since reconstruction in zones around proxies can not offer the same accuracy as reconstruction at proxy locations. Overall, the quality of reconstruction at locations in the vicinity of proxies is lower, but follows the same patterns than the quality of reconstruction at proxy locations.

TS punctual correlations in zones around proxies

In Figure 3.10 we can see the average quality of reconstruction based on correlation on TS data around proxy locations. A first general comment is that the free reconstruction shows almost no degree of correlation for any set of proxy locations.

As far as the TS reconstruction over TS proxy zone (blue) is concerned, as compared to Figure 3.8, the performances of all methods, especially of the Base Method without interpolation and the Cumulative Resampling Method have decreased significantly. This is true for both time series. On the other hand, TS reconstruction over SST proxy zones (red), are much closer to their counterpart in Figure 3.8. One possible hypothesis could be that, as ocean circulation is slower than atmosphere circulation, spatial correlations could be lower with atmospheric data than with ocean data.

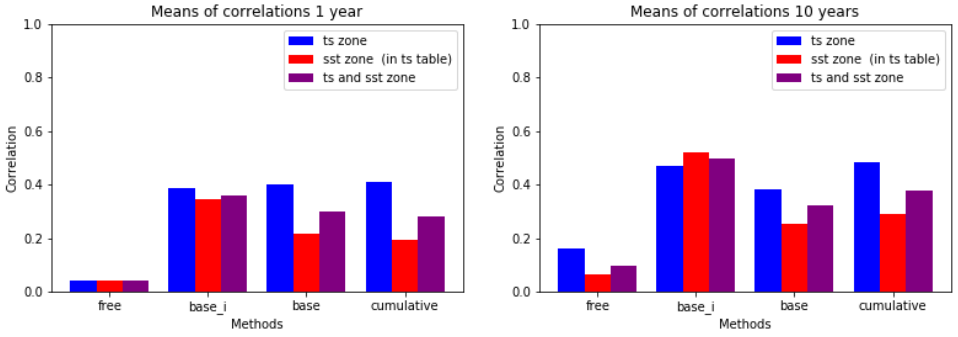

Figure 3.10: TS Correlation Coefficient Averages in the SST and TS proxy zones, 1 year(left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the TS in TS proxy zone (blue), at SST proxy zone (red) and at SST & TS proxy zones (purple), as shown in Figure 3.5. The correlations were computed on two different temperature data series, 1 year averages (left), 10 years averages (right).

Finally, the TS reconstruction over TS and SST proxy zones (purple) show slightly better correlation coefficients for the Base Method with interpolation. Inevitably, this difference is induced by the Base Method with interpolation's greater accuracy on ocean reconstruction.

3.3.4 Correlations between Geographical Temperature Averages

We will now focus on correlations between geographical temperature averages. This allows to determine the quality of the reconstruction on geographical zones in the general vicinity of the proxies. Similar to Section 3.3.3, analysing for zones does not allow comparison with the correlations on the proxies. This is explained by the fact that proxy data is not given over zones containing multiple grid cells. Hence the "proxy" column will not be featured in the bar plots.

SST correlations between zone temperature averages

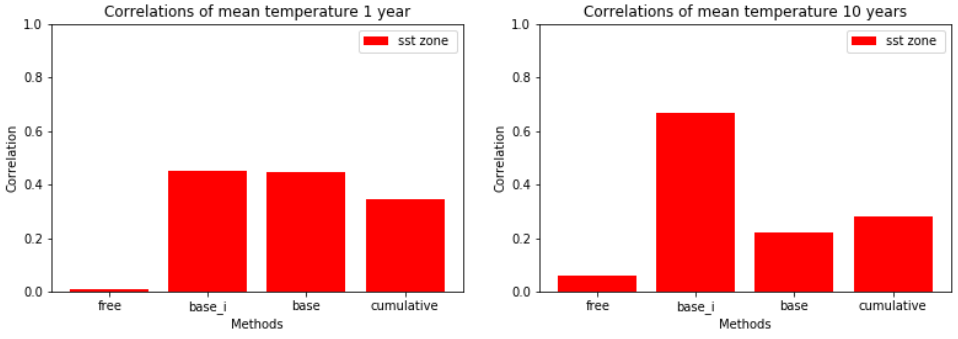

Figure 3.11: Correlation Coefficients over average SST on SST proxy zone, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the correlations on SST averages in the SST proxy zone as shown in Figure 3.4. The correlations were computed on two different temperature data series, 1 year averages (left), 10 years averages (right)

The bar plots in Figure 3.11 show a relatively poor reconstruction of the average temperature on the SST proxy zone. This is especially true for the Base Method without interpolation and the Cumulative Resampling Method which, surprisingly, suffer a lot from an increase in

times series scale. We have no explanation for this evolution. Nevertheless, it can be observed at a 10 year reconstruction that some grid cells showcase an extremely poor correlation in the direct vicinity of proxy locations. These cells may be very impactful to the computation of the average on the entire zone, subsequently influencing the calculation of this correlation.

TS correlations between zone temperature averages

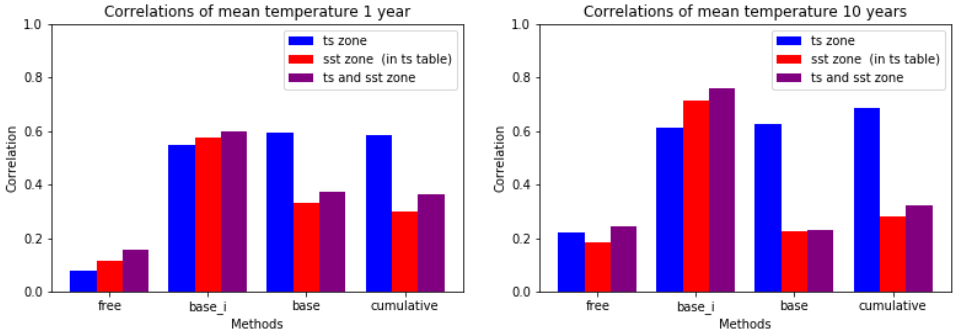

Figure 3.12: Correlation Coefficients over average TS on TS and SST proxy zones, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods (Free Run, Base Method with interpolation, Base Method without interpolation, Cumulative Resampling Method) on the TS averages at TS proxy zone (blue), at SST proxy zone (red) and at SST & TS proxy zones (purple),as shown in Figure 3.5. The correlations were computed on two different temperature data series: 1 year averages (left), 10 years averages (right).

The bar plots in figure 3.12 reinforce arguments for the effectiveness of the Base Method with interpolation even further. It is better in reconstructing the average temperature in combined regions than any other method in this setup. Moreover, the Free Reconstruction seems to also become quite effective on averaging over larger areas, becoming nearly as good as the Base Method and the Cumulative Resampling Method on 10 year timescales.

As far as the TS reconstruction over TS proxy zone (blue) is concerned, as compared to Figure 3.10, the performances of all methods, have increased and are similar in correlation coefficient on all timescales. Considering the TS reconstruction over SST proxy zone (red), one can observe a similar pattern than for SST reconstructions over SST proxy zones (Figure 3.11).

An interesting phenomenon on these graphs is that the correlation coefficient of the reconstruction over the combination of the SST and TS proxy zones (purple) is, with the Base Method with interpolation, higher than for either separate zone. This indicates that the average temperature over the union of the two zones is more strongly correlated with the reference run than either zone is alone. One hypothesis is that deviations from the reference run in the separate zones could be opposite to each other. These (likely random) deviations would balance each other out when averaging over the union of the zones.

TS and SST correlations between northern hemisphere temperature averages

The plots in Figures 3.13 and 3.14 display the quality of reconstruction of each method on respectively SST's and TS's across the northern hemisphere. The first important observation is that the free reconstruction performance on the TS is here much higher in comparison to the other methods. Indeed, climate forcings have more noticeable impacts on the atmosphere on a global scale, which explains this correlation between separate runs. Very interestingly here, especially on the sea surface temperatures (but also on surface temperatures), the Base Method without interpolation and the Cumulative Resampling Method seem to be doing better than the Base Method with interpolation (which had outperformed both in the previous measurements). It could be that similarly to what was observed in Lorenz63, the fact that values of proxies are interpolated pushes the filters into unlikely states. This would

not affect too much the reconstruction on interpolated variables, however all other variables would be forced into unlikely states which may explain this decline in precision.

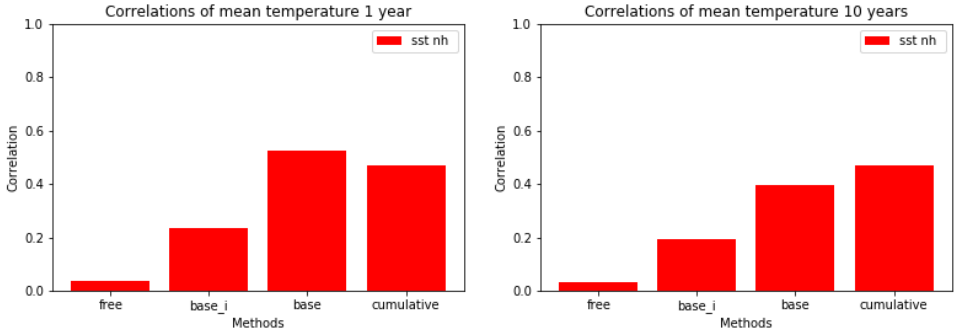

Figure 3.13: Correlation coefficients over average temperature on SST in the northern hemisphere, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods on the SST's in the northern hemisphere. These plots show correlation coefficients over different timescales. On the left we have the yearly correlation coefficient while on the right we have the decadal correlation coefficient.

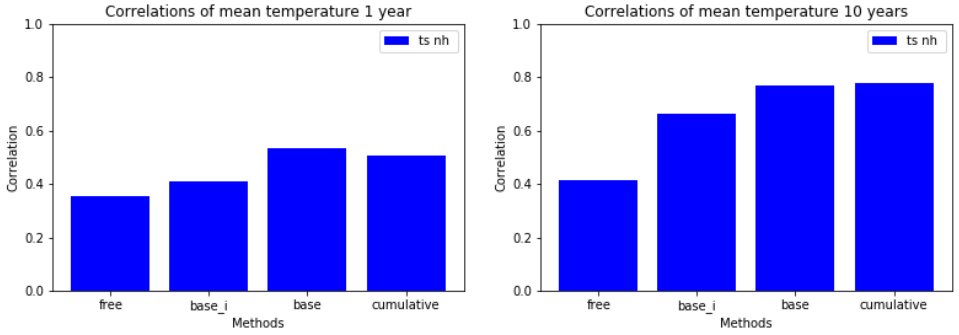

Figure 3.14: Correlation coefficients over average temperature on TS in the northern hemisphere, 1 year (left) and 10 years (right)

These two bar plots illustrate the quality of reconstruction of the different methods on the TS's in the northern hemisphere. These plots show correlation coefficients over different timescales. On the left we have the yearly correlation coefficient while on the right we have the decadal correlation coefficient.

3.3.5 Exploration with the Coefficient of Efficiency

Metrics used to assess our results in this chapter can be flawed. Some, like the time series plots, lack quantitative information. Others, like correlation coefficients, generally give a decent quantification of the results but can also give a false impression of similarity. The ways in which correlations can flaw the perception of similarity is shown in Figure 3.15. At first sight, the Free Reconstruction (left, an average of 10 free runs) does not seem to reconstruct the reference very well, while the Cumulative Resampling Method (right), although the curve is shifted upwards, seems to better follow the evolution of the reference. This impression is contradicted by the correlation coefficients, 0.52 for the Free Reconstruction and 0.36 for the Cumulative Resampling Method. Hence, it can be useful to use an additional metric to evaluate the performance. The Coefficient of Efficiency (CE) can be a helpful candidate.

Drawing conclusions from RMSE measurements on grid points in a climate reconstruction is hard, this is because different parts of the globe are subject to different climate variability (explored in Appendix .3). Nevertheless, if we were to divide the RMSE of the variables in each grid point with their variances, we might obtain a useful indication on the quality

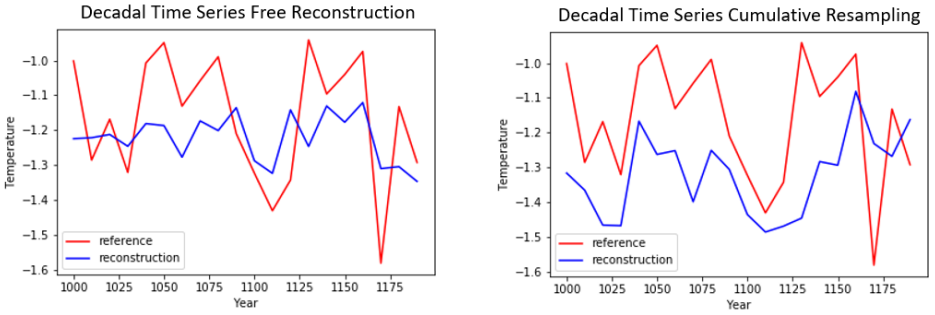

Figure 3.15: TS decadal time series over the zone enclosing all proxies, Free Reconstructions averages (left), Cumulative Resampling Method (right)

These decadal TS time series plots on the period of interest allow us to compare the reconstructions of different methods with the reference run. The Free Reconstruction on the left shows the average TS of all 10 free runs. The Cumulative Resampling reconstruction on the right shows the average TS of the reconstructed trajectory. The correlation on the time series is of 0.52 on the left and 0.36 on the right.

of reconstruction. This is the idea behind the coefficient of efficiency. The formula for the coefficient of efficiency (taken from [14]) of a time series of length l is given as:

$$CE = 1 - \frac{\sum_{t=0}^{l-1} (x_t - \hat{x}_t)^2}{\sum_{t=0}^{l-1} (x_t - \overline{x}_t)^2} \tag{3.1}$$

Where x gives the reference time series, $\hat{x}$ the reconstructed time series and $\overline{x}$ the average of the reference time series. If we were to divide both the numerator and denominator by l in the fraction term in Equation 3.1, it would become the MSE (mean square error) of $\hat{x}$ divided by the variance of x. This fraction is subtracted from 1 to give an idea on the usefulness of the reconstruction. If the coefficient of efficiency is equal to 1, the reconstruction is perfect. If the coefficient is between 0 and 1, the reconstruction has some errors but these remain inferior to the natural variance on x, therefore the assimilation is still relevant. If the coefficient is 0 or less, we are better off just taking the average on $\hat{x}$ as a constant for the reconstruction. In that case, the assimilation becomes mostly irrelevant.

Figures 3.16 and 3.17 were included to compare the correlation coefficients and the coefficients of efficiency. As can be observed, geographical patterns on coefficients of efficiency mirror the geographical patterns on correlation coefficients quite well. Although not visible on the coefficient of efficiency graph (because it would ruin the visibility if we were to bring the colour spectrum so low), the main difference between coefficient of efficiency and correlation coefficient lies in their extremes. The lowest value of the correlation coefficient in Figure 3.16 is of -0.23 while the lowest value for the coefficient of efficiency in Figure 3.17 is of -8.60. This would influence the measures of average performance in zones, as explored in Section 3.3.2.

To sum up, it is important to keep in mind that correlation is not necessarily the best metric in measuring the reconstruction quality. In order to have a fuller picture, one can look at all the maps on the different metrics included in Appendix .4.

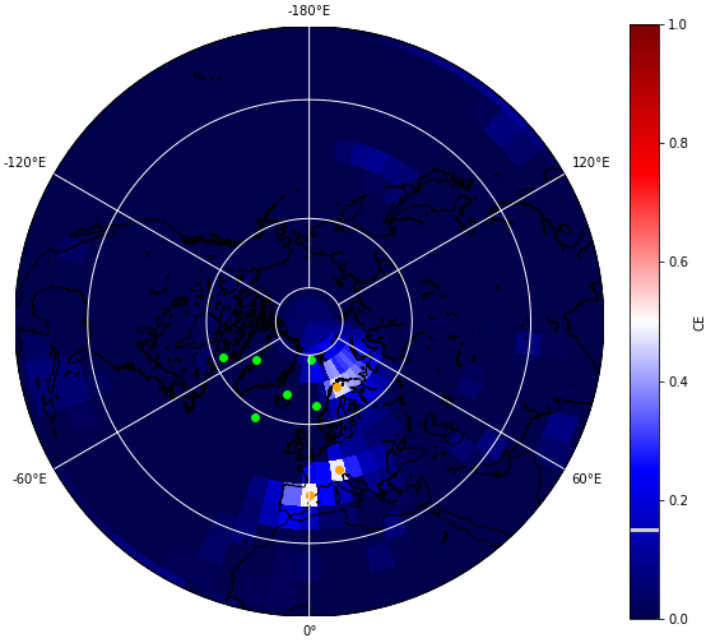

Figure 3.16: Illustration of yearly Correlation Coefficient on TS

This plot of the northern hemisphere in a North stereographic projection shows the Coefficient of Correlation of the Base Method without interpolation reconstruction on TS values in the cells of the ECBilt grid.

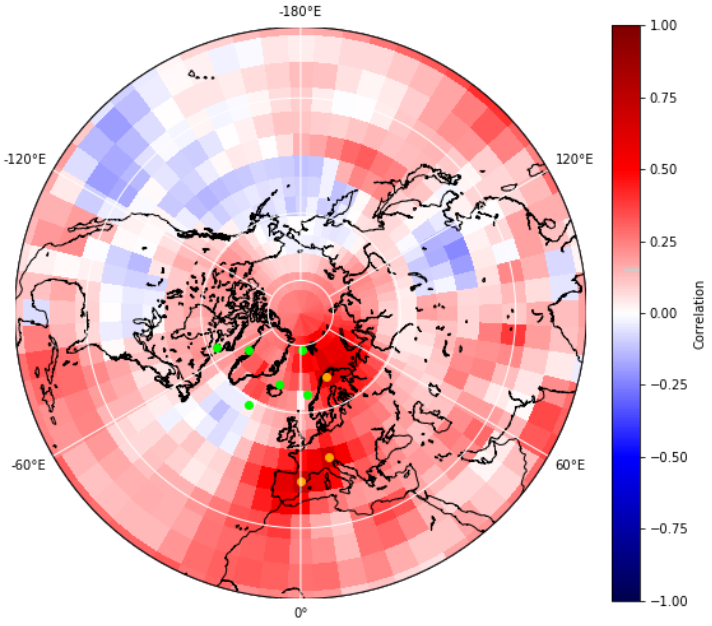

Figure 3.17: Illustration of yearly CE on TS

This plot of the northern hemisphere in a North stereographic projection shows the Coefficient of Efficiency of the Base Method without interpolation reconstruction on TS values in the cells of the ECBilt grid.

3.3.6 Comments on Method Degeneracy

As explored in Lorenz63, some methods are more prone to degeneracy than others. While it could sometimes be affected by the method ability to follow the reference trajectory (the worse the reconstruction of method, the more it tended to be degenerated), it could also give an idea about the scalability of the methods. In order to visualise how much the methods were actually prone to degeneracy on LOVECLIM, we calculated the average minimum amount of particles required to represent a certain percentage of the total weight. The following results were obtained:

	Base Method without interpolation	**Base Method with interpolation**	**Cumulative Resampling**
50%	12.56	2.84	1.45
90%	35.64	10.69	2.95

Table 3.3: Minimum Number of Particles for 50% and 90% of the Weight

As can be seen on Table 3.3, the method which is the most prone to degeneracy is the Cumulative Resampling Method, with over 50% of the weight contained on average on 1.45 particles which, out of 96, is a very small number. The Base Method with interpolation also has a dense weight distribution in the heaviest particles, but it still has around 10 particles sharing at least 90% of the weight. Finally, the Base Method without Interpolation seems to be the least prone to degeneracy here, with 12.56 and 35.64 particles on average to clump respectively 50% and 90% of the weight.

Chapter 4

Conclusion

Paleoclimatology makes use of particle filters to reconstruct past climates. They do so with the help of climate proxies (climate archives found in nature such as corals, tree rings, etc...). These proxy records come with a wide variety of time resolutions. Hence particle filters, which work on a single timescale, have trouble to adapt to these varying timescales. In the light of this shortcoming, we sought to determine how to reconstruct the state of a system when observations were made on two variables which were available over two different timescales. We did so with the help of some well established and novel online particle filtering methods. Our two main objectives were to assess the filters' accuracy and resistance to degeneration. The two established particle filters used as benchmarks being the Base Method (a regular SIR particle filter) with and without interpolation and the investigated particle filters being:

- The Particle Backtracking Method: resampling at high frequency, designed to provide a better reconstruction and a high resistance to degeneracy.
- The Cumulative Resampling Method: resampling at low frequency, designed for a very precise reconstruction with a possible increased risk of degeneracy.
- The Conditional Resampling Method: resampling at variable frequency, designed for a high reconstruction precision and a high resistance to degeneracy. It can be seen as a combination between Particle Backtracking and Cumulative Resampling.

In order to evaluate the performance of the methods in versatile situations with little computational requirement, the methods were tested on multivariate setups on Lorenz63, a three dimensional chaotic system. We wanted to assess whether the investigated methods could be scalable to more complex climate models (LOVECLIM). RMSE and correlation analyses were performed. The Base Method without interpolation was chosen as the benchmark method. The performance evaluations on the different methods were as follows:

- The Particle Backtracking method's performance was quite good. In fact, it was never lower than the Base Method's and was robust against degeneracy. This was promising regarding its scalability to LOVECLIM.
- The Cumulative Resampling acted as a double edged sword. On one hand, it was highly efficient at reconstructing big portions of the simulation when the particle number was unreasonably high. On the other hand, it struggled with degeneracy when the particle number was low. This shed doubts on its scalability to LOVECLIM.
- Conditional Resampling offered hopeful results with a condition on entropy of the likelihood vector. Not only could it, with the proper fine tuning, have the best of both worlds between its parent methods (the robustness from Particle Backtracking and the precision from Cumulative Resampling), it could also reconstruct better than both of them in some setups.

In the light of these results, all methods were elected for application to LOVECLIM. In this setting we analysed Both Base Methods and the Cumulative Resampling method (at the time of submission of this document, the reconstructions for the Particle Backtracking were still ongoing). We decided to opt for a pre-industrial climate reconstruction with proxies (6

ocean decadal proxies, 3 land yearly proxies) located in the upper northern hemisphere and concentrated around the eastern Atlantic. We performed correlation analyses to evaluate reconstructions in different zones on and around the proxy locations.

The methods worked on LOVECLIM somewhat differently than what could have been expected from Lorenz63 testing. The Base Method without interpolation showed the lowest degree of degeneracy and maintained a good assimilation on land proxies. It also showed a decent reconstruction around the ocean proxies although not taking them in consideration. On the other hand, the reconstruction of the Base Method with interpolation, which did poorly on Lorenz63 compared to the other methods, performed comparatively very well on LOVECLIM. This was likely due to its prioritisation on ocean proxies which made its zone reconstructions much better. Furthermore, the Cumulative Resampling Method, which was expected to go badly on a more complex system like LOVECLIM, showed acceptable results. Although being at the verge of degeneracy, it showcased good reconstruction on land proxies, while not being as efficient on ocean proxies.

These results show that all investigated methods can be applied to both systems with some limitations. This has yet to be established for the Particle Backtracking and the Conditional Resampling Methods. We are looking forward to these results and are confident that they will be of great interest. With these considerations, we hope that the methods explored in this document will provide some food for thought as to potentially new exploration grounds on data assimilation.

Bibliography

[1] Gregory J. Hakim, J. Annan, S. Brönnimann, M. Crucifix, T. Edwards, H. Goosse, A. Paul, G. van der Schrier and M. Widmann "Overview of data assimilation methods" *Science Highlights: Data assimilation*, accessed 22-06-2021 https://boris.unibe.ch/39169/1/PAGESnews_2013(2)_72-73_Hakim.pdf

[2] "What Are 'Proxy' Data?" *National Climatic Data Centre*, accessed 22-06-2021 www.ncdc.noaa.gov/news/what-are-proxy-data.

[3] Goosse, H., Brovkin, V., Fichefet, et al.: *Description of the Earth system model of intermediate complexity LOVECLIM* version 1.2, Geosci. Model Dev., 3, 603–633, https://doi.org/10.5194/gmd-3-603-2010, 2010

[4] Peter Jan van Leeuwen *Particle Filtering in Geophysical Systems* University of Reading, 2009

[5] Davis, J.C. *Statistics and Data Analysis in Geology.* Wiley International Edition, John Wiley Sons, Inc., 1973,New York, 550 pp.

[6] Steven W. Smith *The Scientist and Engineer's Guide to Digital Signal Processing*, Chapter 17: Recursive Filters,1997

[7] Evensen, G. *The Ensemble Kalman Filter: theoretical formulation and practical implementation.* Ocean Dynamics 53, 343–367 (2003). https://doi.org/10.1007/s10236-003-0036-9

[8] F. Daum, *Nonlinear filters: beyond the Kalman filter* in IEEE Aerospace and Electronic Systems Magazine, vol. 20, no. 8, pp. 57-69, Aug. 2005, doi: 10.1109/MAES.2005.1499276

[9] F. Daum and J. Huang, *Curse of dimensionality and particle filters* 2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652), 2003, pp. 4_1979-4_1993, doi: 10.1109/AERO.2003.1235126.

[10] G. A. Kivman *Sequential parameter estimation for stochastic systems* Alfred Wegener Institute for Polar and Marine Research, Bremerhaven, Germany, 2002

[11] L. Slivinsky, C.Snyder, 2015: *Exploring Practical Estimates of the Ensemble Size Necessary for Particle Filters*

[12] Snyder, C., T. Bengtsson, and M. Morzfeld, 2015: *Performance bounds on particle filters using the optimal proposal.* Mon. Wea. Rev., 143, 4750–4761

[13] Dubinkina, S., Goosse, H., Sallaz-Damaz, Y., Crespin, E., & Crucifix, M. (2011). *Testing a particle filter to reconstruct climate changes over the past centuries. International Journal of Bifurcation and Chaos*, 21(12), 3611-3618. https://doi.org/10.1142/S0218127411030763

[14] Steiger, N. and Hakim, G.: *Multi-timescale data assimilation for atmosphere–ocean state estimates*, Clim. Past, 12, 1375–1388, https://doi.org/10.5194/cp-12-1375-2016, 2016.

[15] K. J. H. Law, A. Shukla, A. M. Stuart: *Analysis of the 3DVAR Filter for the Partially Observed Lorenz '63 Model* Warwick Mathematics Institute, 2013

[16] C. Robert, G. Casella: *Monte Carlo Statistical Methods*, Springer-Verlach, 2004

[17] H. Goosse: *Introduction to climate dynamics and climate modelling*, Cambridge University Press, 2015

[18] University Corporation for Atmospheric Research: *The Lorenz 63 model and its relevance to data assimilation*, 2021, Revision f3f100e5, https://docs.dart.ucar.edu/en/latest/guide/lorenz-63-model.html

[19] Jennifer Flannery: *Coral Reefs as Climate Archives*, St. Petersburg Coastal and Marine Science Center, seen on 20 October 2021 https://www.usgs.gov/centers/spcmsc/science/coral-reefs-climate-archives?qt-science-center-objects=0qt-science-center-objects

[20] National Center for Environmental Information *Picture Climate: How Can We Learn from Tree Rings?*, seen on 20 October 2021 https://www.ncdc.noaa.gov/news/picture-climate-how-can-we-learn-tree-rings

[21] Mark R. Chapman and Nicholas J. Shackleton, *What level of resolution is attainable in a deep-sea core? Results of a spectrophotometer study* Godwin Institute for Quaternary Research, 1998

[22] Tiancheng Li, Miodrag Bolic´, and Petar M. Djuric´, *Resampling Methods for Particle Filtering; Classification, implementation, and strategies* IEEE Signal Processing Magazine, 2015

[23] Evensen, G. *The Ensemble Kalman Filter: theoretical formulation and practical implementation.* Ocean Dynamics 53, 343–367 (2003). https://doi.org/10.1007/s10236-003-0036-9

[24] Edward N. Lorenz, *Deterministic Nonperiodic Flow*, Massachusetts Institute of Technology, 1963

[25] Arab Djebbar, Hugues Goosse and François Klein, *Robustness of the Link Between Precipitation in North Africa and Standard Modes of Atmospheric Variability During the Last Millennium*, Climate, 2020 doi: 10.3390/cli8050062

[26] Delcourt, François. *Particle filters using sparse observations : applications in palaeoclimatology.* Ecole polytechnique de Louvain, Université catholique de Louvain, 2017. Prom. : Absil, Pierre-Antoine ; Goosse, Hugues. http://hdl.handle.net/2078.1/thesis:12868

[27] Corrado Gini, *On the Measure of Concentration with Special Reference to Income and Statistics*, Colorado College Publication, 1936

[28] Wikipedia, Gini coefficient, https://en.wikipedia.org/wiki/Gini_coefficient, viewed on 19 November 2021

[29] Jan Esper et al., Orbital forcing of tree-ring data, Nature Climate Change, 2012, doi:10.1038/nclimate1589

[30] U. Bütgen et al., 2500 Years of European Climate Variability and Human Susceptibility , Science, 2011, doi: 10.1126/science.1197175

[31] I. Dorado Liñán et al, *Estimating 750 years of temperature variations and uncertainties in the Pyrenees by tree-ring reconstructions and climate simulations*, 2012, doi: 10.5194/cp-8-919-2012

[32] Sicre et al, *Sea surface temperature variability in the subpolar Atlantic over the last two millennia*, 2011, doi:10.1029/2011PA002169

[33] Calvo et al, High resolution U37K sea surface temperature reconstruction in the Norwegian Sea during the Holocene,Quaternary Science Reviews, 2002, doi:10.1016/S0277-3791(01)00096-8

[34] de Vernal et al,*Dinocyst-based reconstructions of sea ice cover concentration during the Holocene in the Arctic Ocean, the northern North Atlantic Ocean and its adjacent seas*, Quaternary Science Reviews, 2013, doi:10.1016/j.quascirev.2013.07.006G

[35] Bonnet et al.,Variability of sea-surface temperature and sea-ice cover in the Fram Strait over the last two millennia, 2010, doi:10.1016/j.marmicro.2009.12.001

[36] Krawczyk et al., Quantitative reconstruction of Holocene sea ice and sea surface temperature off West Greenland from the first regional diatom data set., Paleoceanography, 2017, doi:10.1002/2016PA003003

[37] WEATHER BLOGS / GLOBAL CLIMATE CHANGE, *What are Climate Forcings?*, Aug. 22 2011, consulted on Dec. 28 2021, https://www.accuweather.com/en/weather-blogs/climatechange/what-are-climate-forcings/48959

Appendices

.1 Application and Visualisation of Particle Backtracking

We want to use the particle data to reconstruct the approximate trajectory of the system, we have to make sure that the mother daughter particle link is preserved at high frequency, this way we get to do our calculations on continues trajectories at high frequency.

The diagram below show how it is possible to maintain the continuous high frequency trajectories to get our reconstruction. The data structure in which the particles are kept is a table which registers the state of the particles at each timestep. The diagram 1 is an example of a timestep assimilation on variable y and 12 timesteps assimilation on variable x. The color code {red, yellow, blue, green} corresponds respectively to the simulated trajectories of particles at indices [0,1,2,3] on a portion of a cycle. The color blending into it indicates the index of the mother particle.

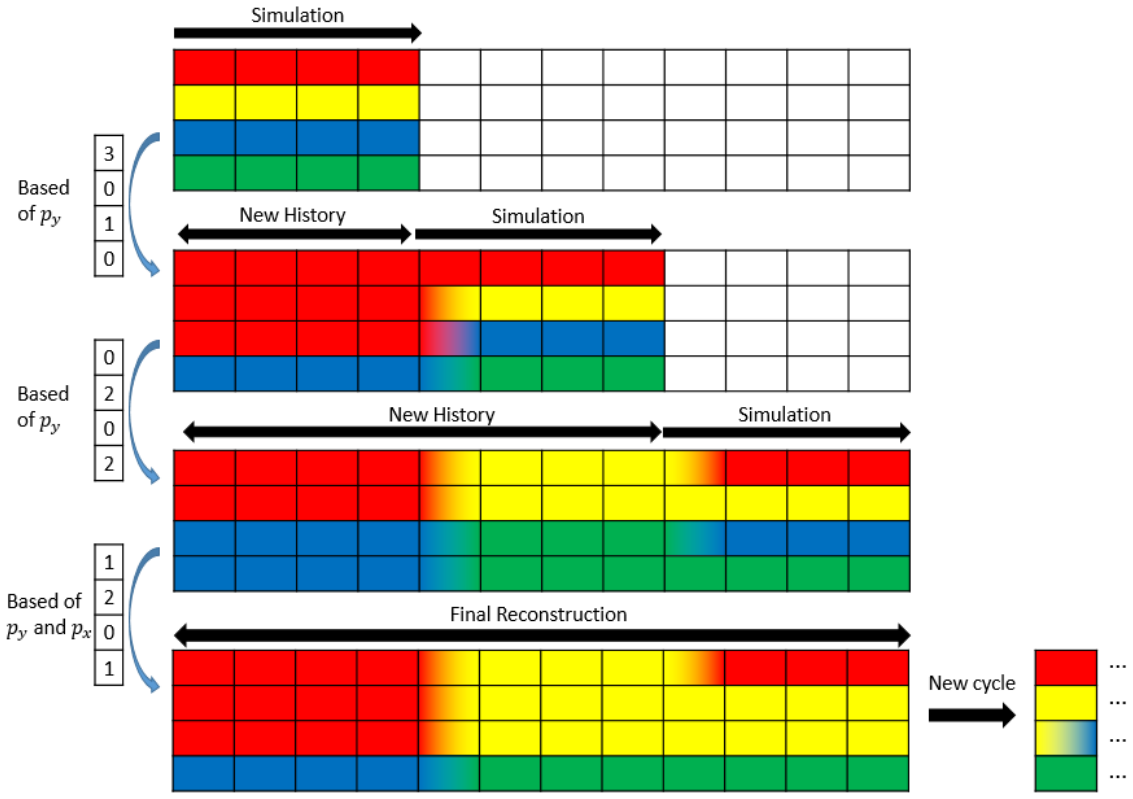

Figure 1: Backtracking Method, manipulation of particle table

.2 Some Attempts Using Gini Coefficient in Conditional Resampling

In economics, the Gini index is used to measure the income or wealth inequality in a country or group [27]. It ranges from 0 to 1 and quantifies how much its wealth distribution differs from a fair distribution. This is mathematically computed through the use of a Lorenz curve. The Lorenz curve is a plot which takes as input the bottom percentile of a population and outputs its proportion of total income. The Gini index measures the ratio between the area under this curve and the "perfect" wealth distribution curve. This is properly illustrated by the Figure 2 (taken from [28]), the Gini index is the ratio of area A over the total area $A+B$.

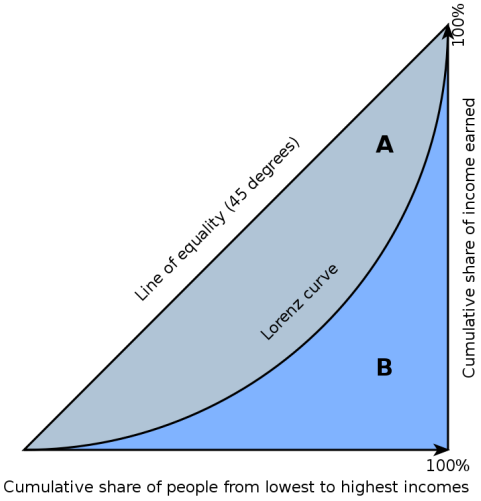

Figure 2: Gini index components [28]

This index could come in handy with the evaluation of weight distribution among the particle likelihoods. Degenerate samples could have a more unequal weight distribution, hence making them prone for resampling, and this should be reflected in their Gini index. In order to better visualise how this index could be of use, we can graph the average Gini coefficients at resampling for the cumulative resampling method in each case assimilating on Lorenz63. In a discrete distribution X such as ours, a gini coefficient can be calculated as follows:

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2n^2\bar{x}} \tag{1}$$

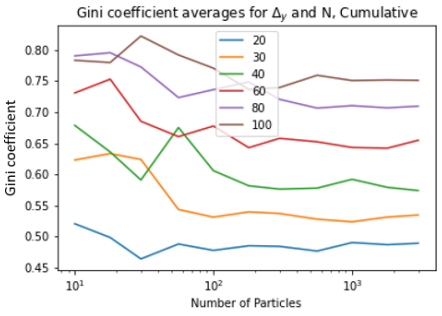

Figure 3: Average Gini, $\Delta_x = 10dt$

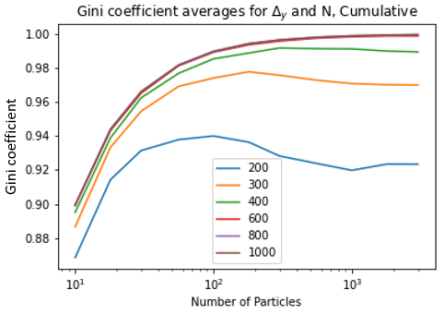

Figure 4: Average Gini $\Delta_x = 100dt$

Unfortunately, it seems like the Gini coefficient is a poor index for this situation since it does not seem to change significantly nor systematically with the number of particles. Thus making it hard to devise a method in which the resampling scheme could toggle between cumulative resampling and particle Backtracking.

.3 RMSE on LOVECLIM

RMSE reconstruction on LOVECLIM was not the best metric to measure the quality of the reconstruction for one main reason. That reason being the annual variability of temperatures in different regions due to the physics of the system. This makes map visualisations very hard. For instance, Figure 5 is the map of the RMSE calculated in each cell by taking the temperature in the reference run and comparing it with the temperature of this cell in the reconstruction. It seems as though the reconstruction is much worse in the polar region because the RMSE is much higher there. This contradicts the results of our correlation assimilation, which showed a better reconstruction in these areas of the globe. Hence using the RMSE is not the correct way of assessing this result. With Lorenz, we were faced with a three dimensional system where variables had a similar range of possible values in magnitude. Here, with a climate simulation, we are bound to have areas with different climate variability from one place to the other, due to the way climate dynamics are, and the way the climate model is programmed. Indeed, when we look at the standard deviation on the temperatures in the LOVECLIM model grid, over a 200 year random simulation we get the following yearly temperature standard deviation (Figure 6).

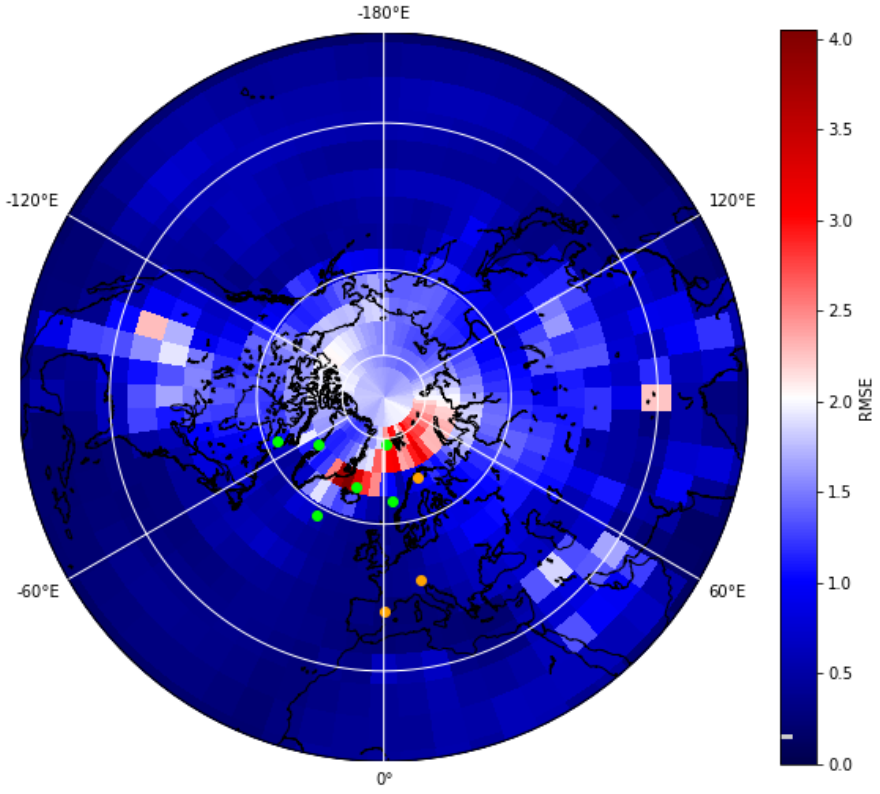

Figure 5: Illustration of the atmospheric RMSE on temperatures in the northern hemisphere

This plot of the northern hemisphere in a North stereographic projection shows the LOVECLIM atmosphere quality of reconstruction with the RMSE metric. If the RMSE is high, there is a higher yearly difference from the reconstruction run to the reference run.

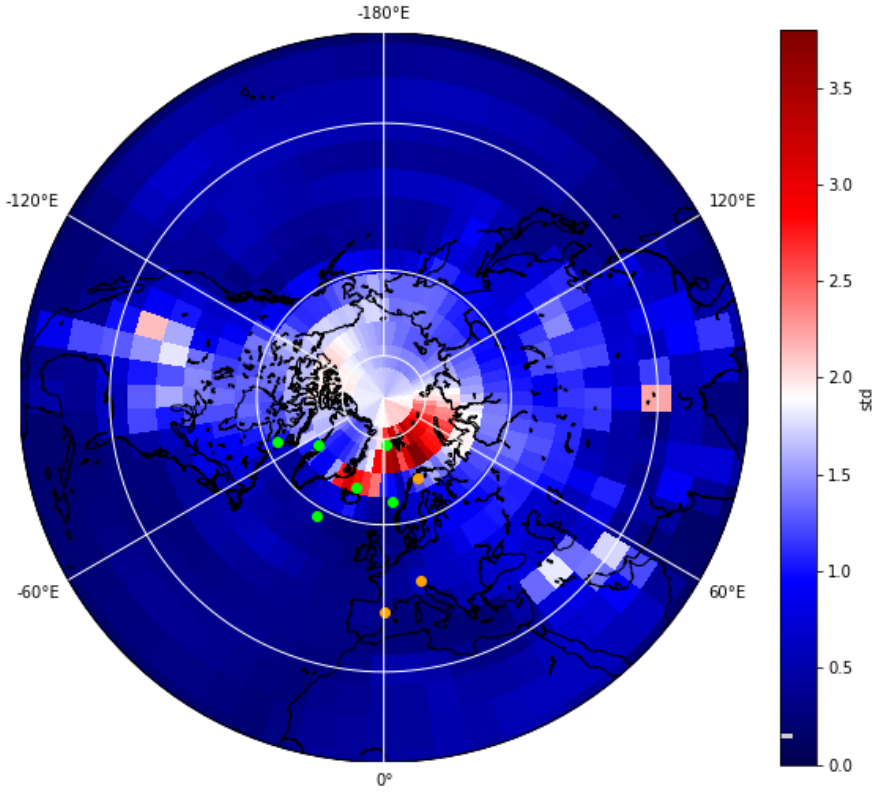

Figure 6: Illustration of the atmospheric yearly temperature standard deviation

This plot of the northern hemisphere in a North stereographic projection shows the yearly temperature distribution standard deviation in the ECBilt grid. If the standard deviation is high, there is a higher yearly average difference between the sample years ones.

First, it is important to point out that these two Figures (5 and 6) do not show the same thing. The comparison is relevant still because the RMSE represents the standard deviation of residuals. In reconstructing a stable climate, RMSE will be strongly linked to the standard deviation of each variable. In all cases, it is clear that RMSE does not allow for an easy visualisation of the quality of reconstruction, due to different regions having much bigger inter-annual variability than others. Hence the reason why we decided to stray away from that analysis entirely.

.4 List of Maps and Reconstructions

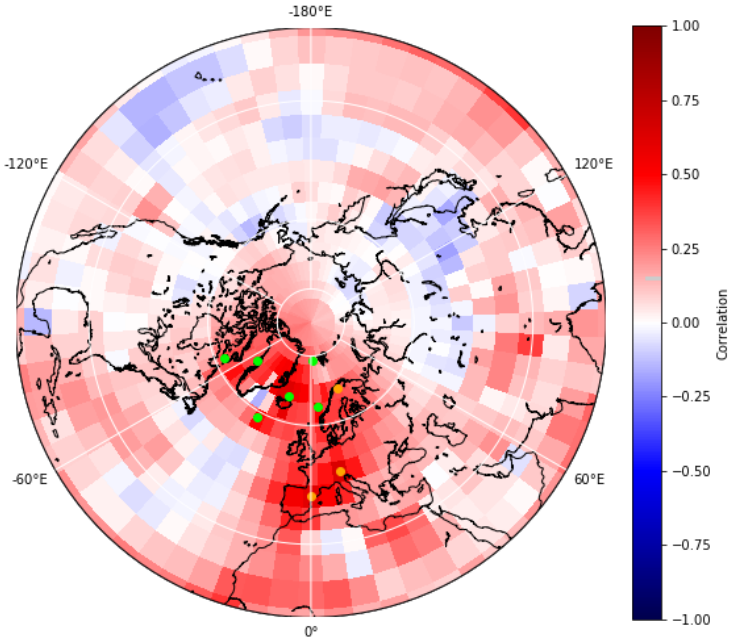

Figure 7: Illustration of yearly Correlation Coefficients on TS, base with interpolation

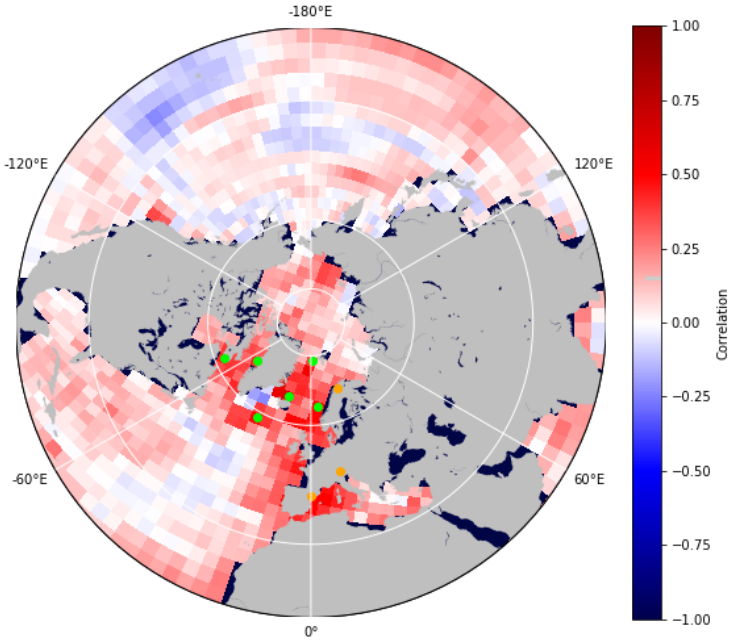

Figure 8: Illustration of yearly Correlation Coefficients on SST, base with interpolation

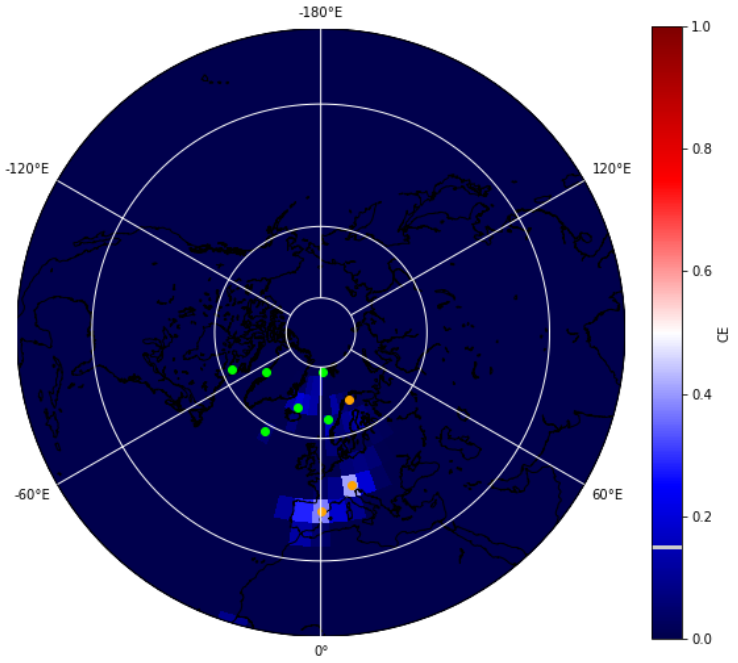

Figure 9: Illustration of yearly CE on TS, base with interpolation

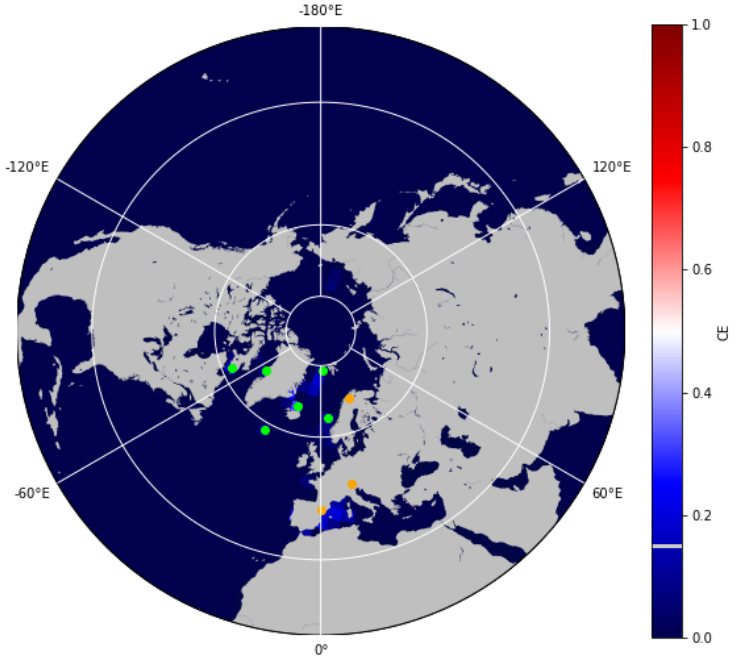

Figure 10: Illustration of yearly CE on SST, base with interpolation

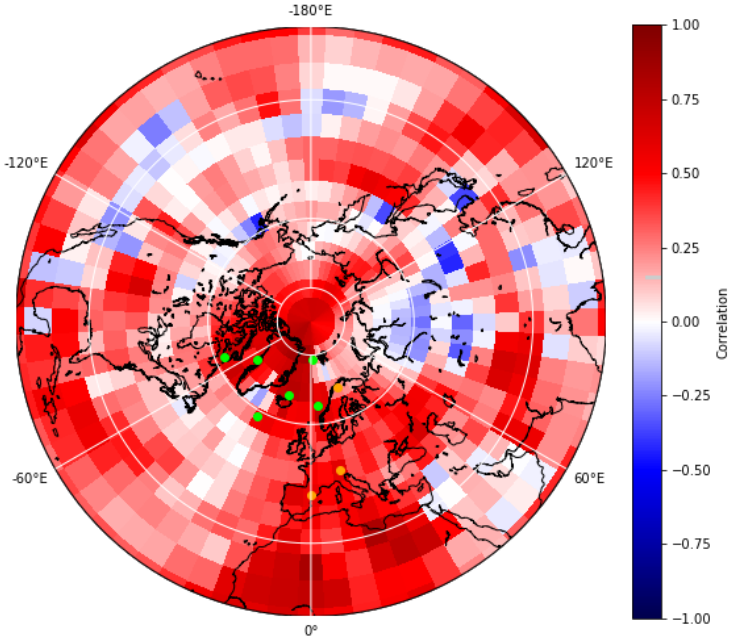

Figure 11: Illustration of decadal Correlation Coefficients on TS, base with interpolation

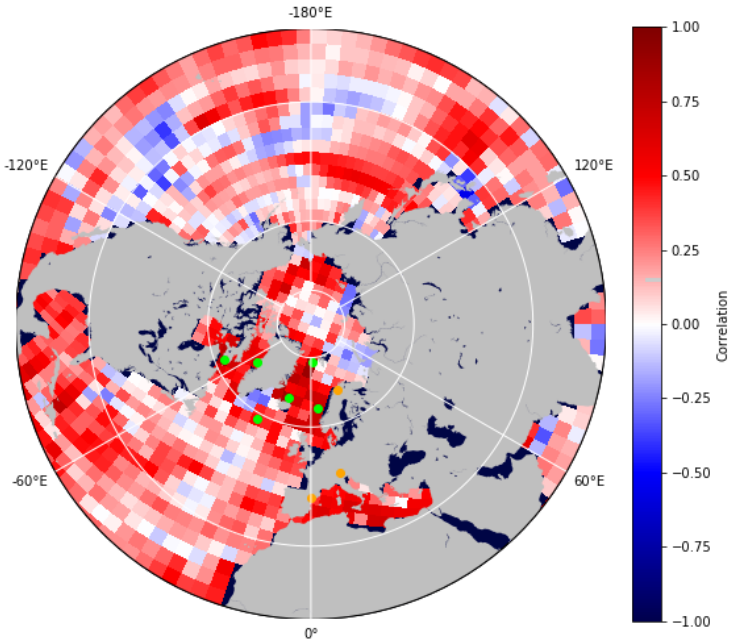

Figure 12: Illustration of decadal Correlation Coefficients on SST, base with interpolation

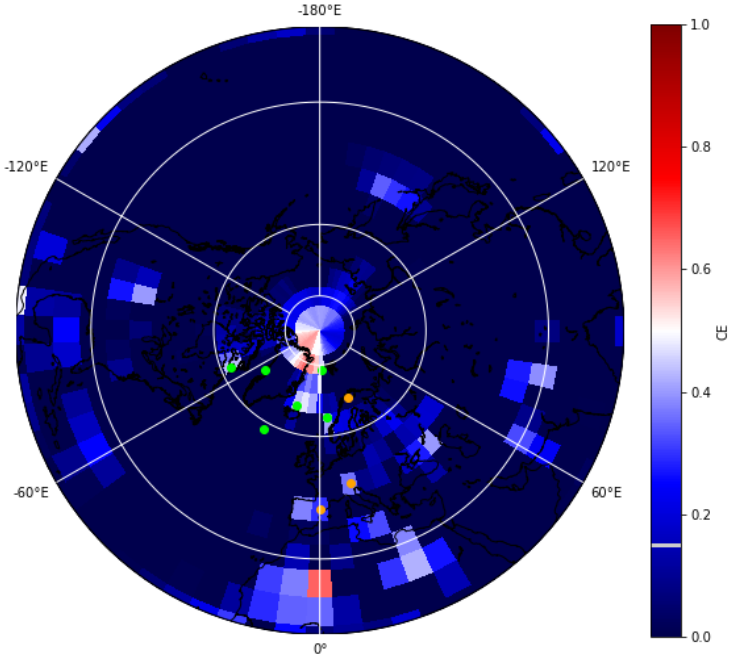

Figure 13: Illustration of decadal CE on TS, base with interpolation

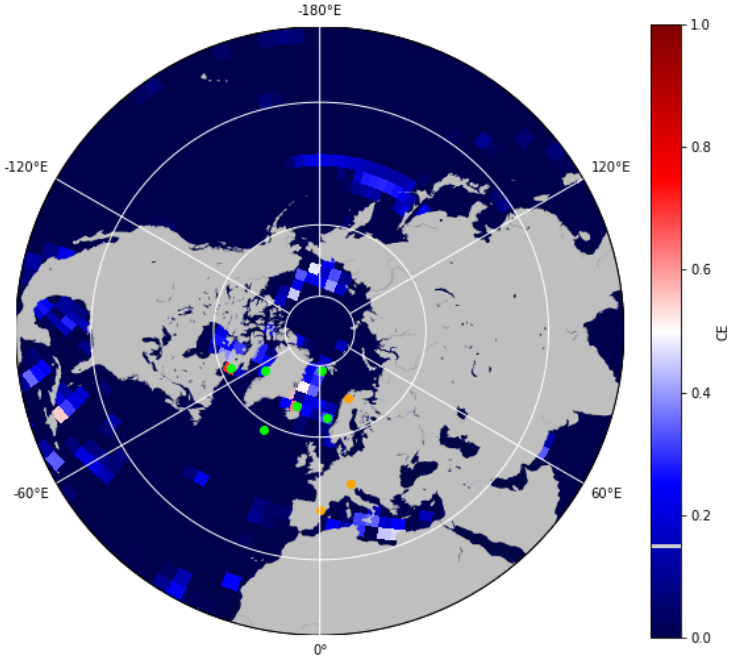

Figure 14: Illustration of decadal CE on SST, base with interpolation

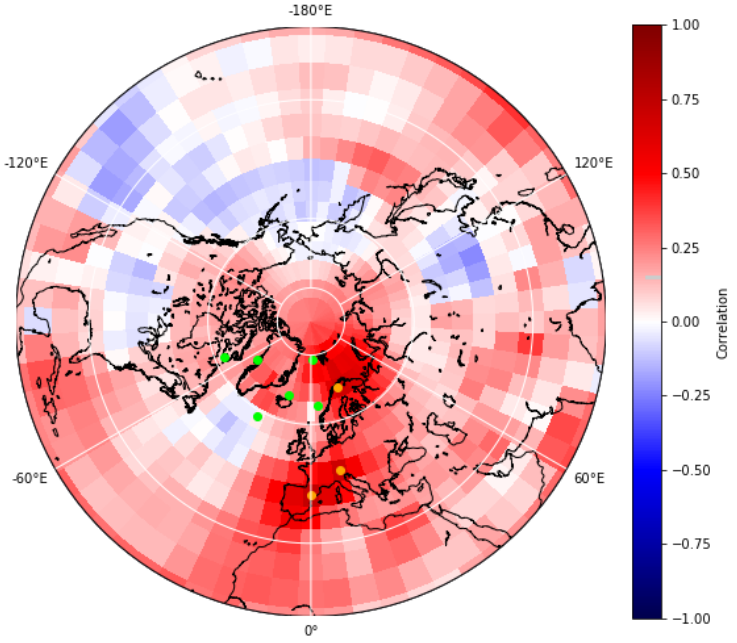

Figure 15: Illustration of yearly Correlation Coefficients on TS, base without interpolation

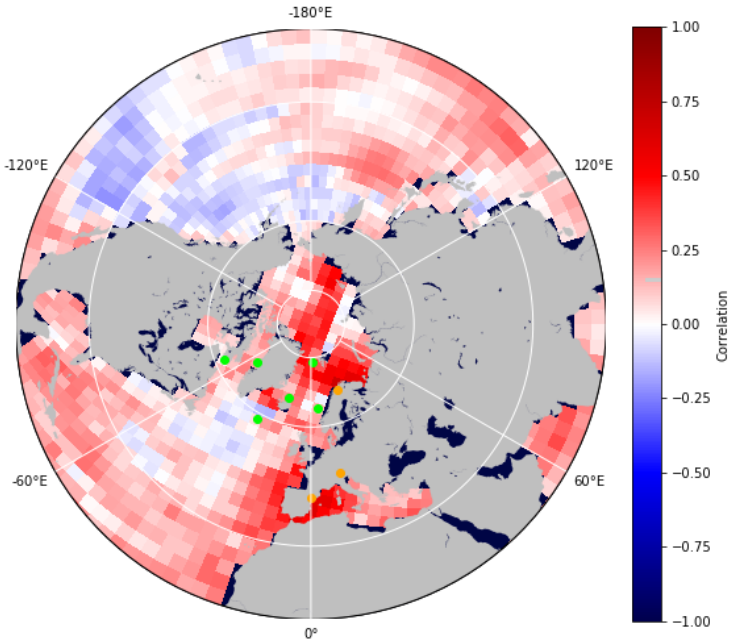

Figure 16: Illustration of yearly Correlation Coefficients on SST, base without interpolation

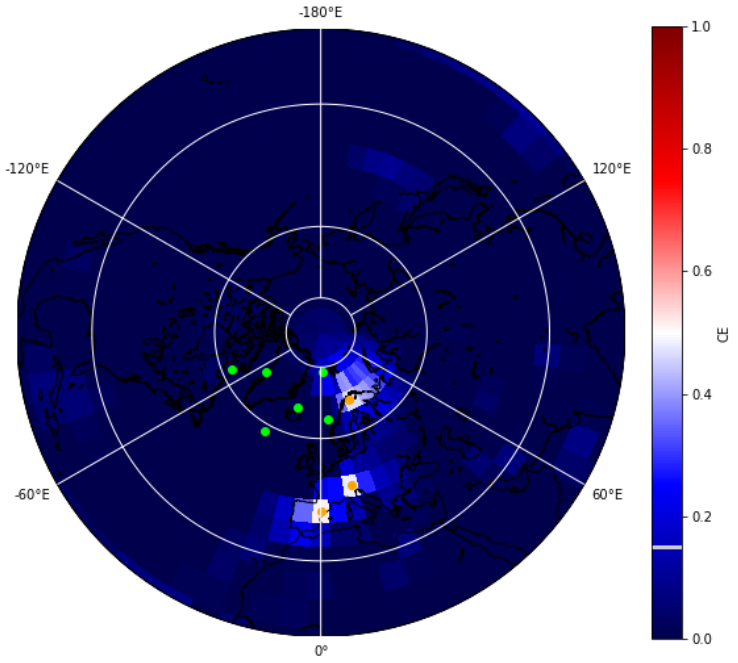

Figure 17: Illustration of yearly CE on TS, base without interpolation

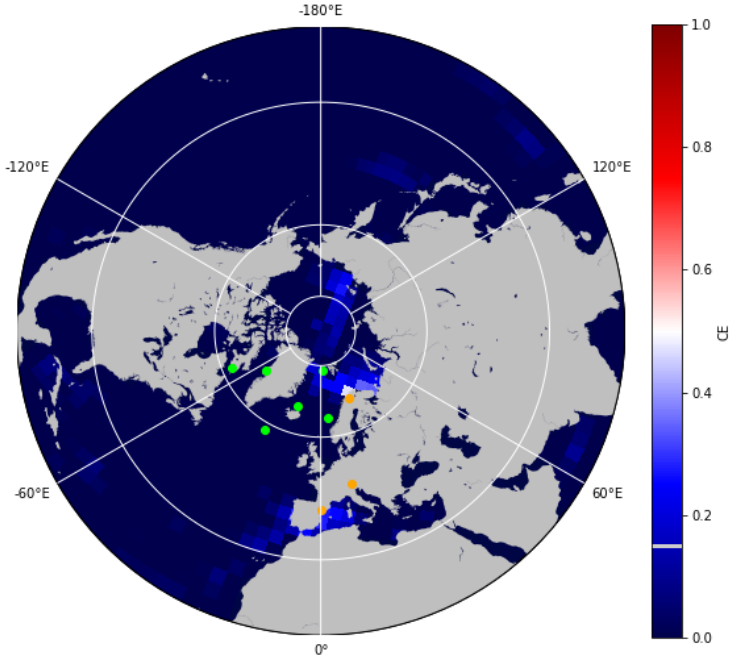

Figure 18: Illustration of yearly CE on SST, base without interpolation

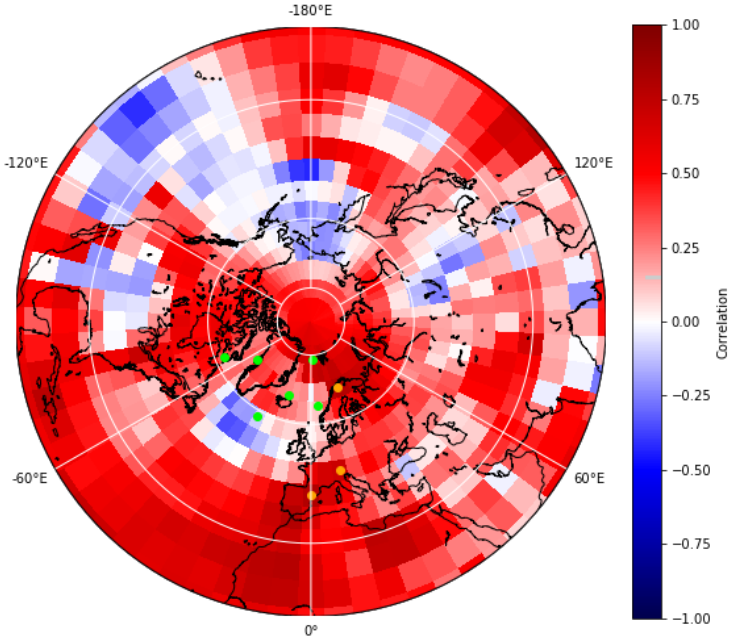

Figure 19: Illustration of decadal Correlation Coefficients on TS, base without interpolation

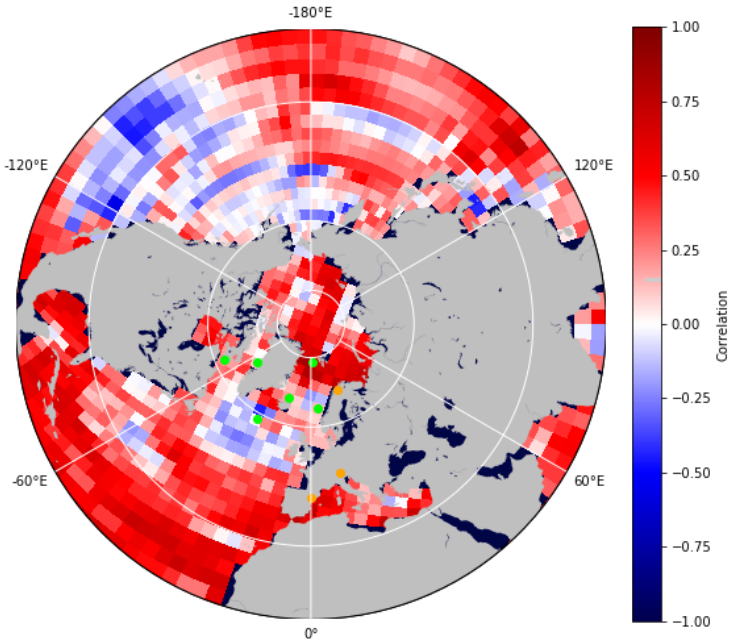

Figure 20: Illustration of decadal Correlation Coefficients on SST, base without interpolation

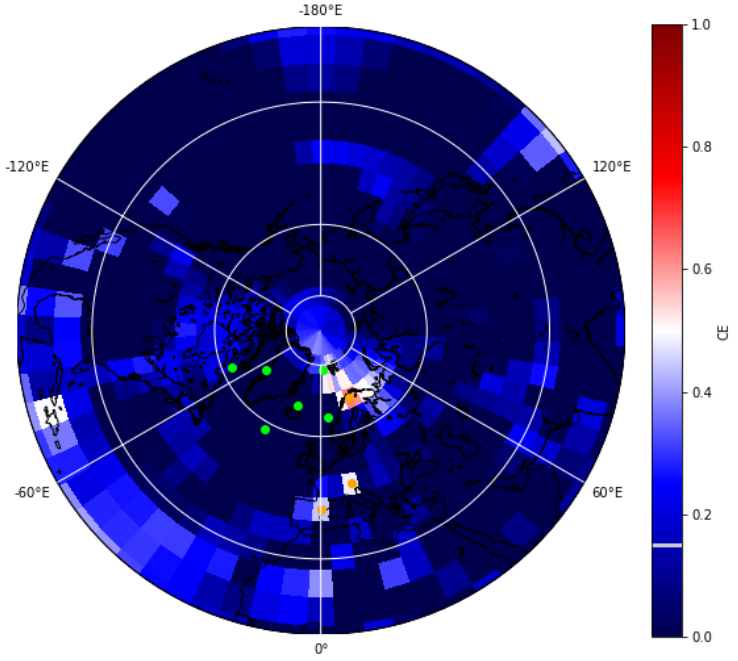

Figure 21: Illustration of decadal CE on TS, base without interpolation

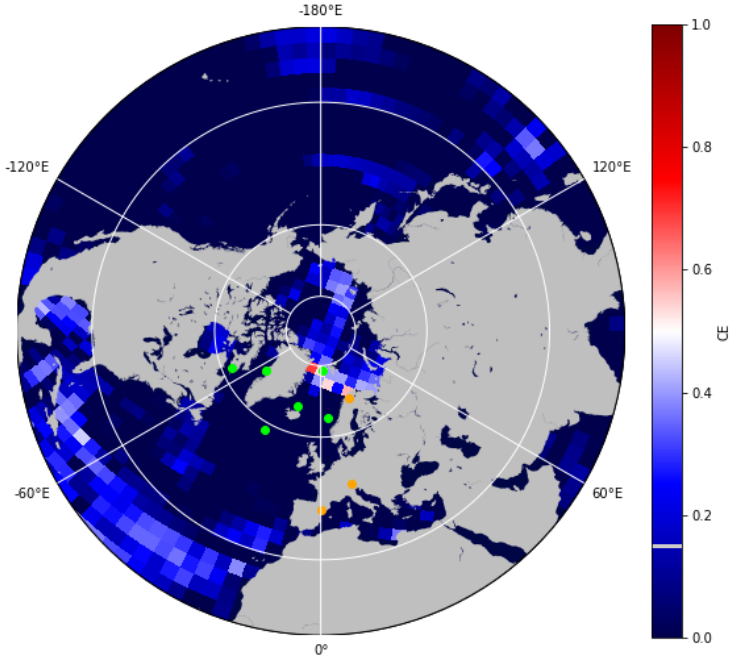

Figure 22: Illustration of decadal CE on SST, base without interpolation

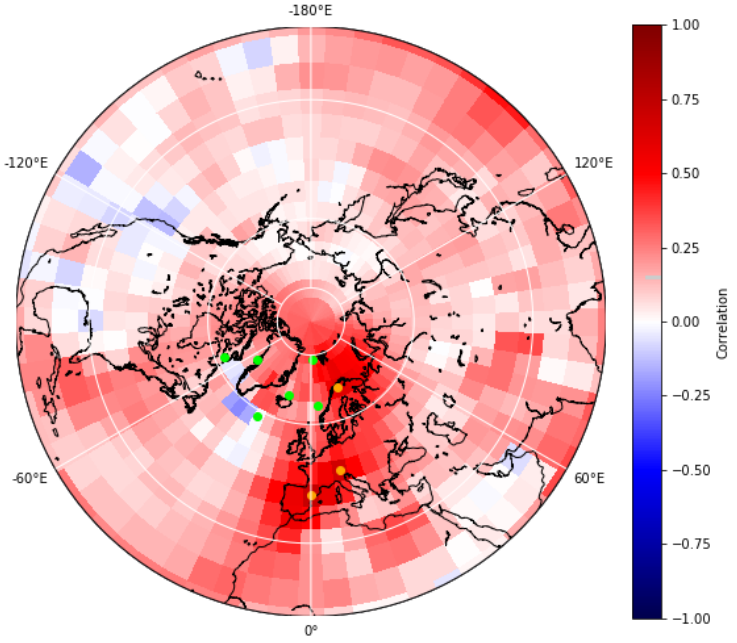

Figure 23: Illustration of yearly Correlation Coefficients on TS, cumulative

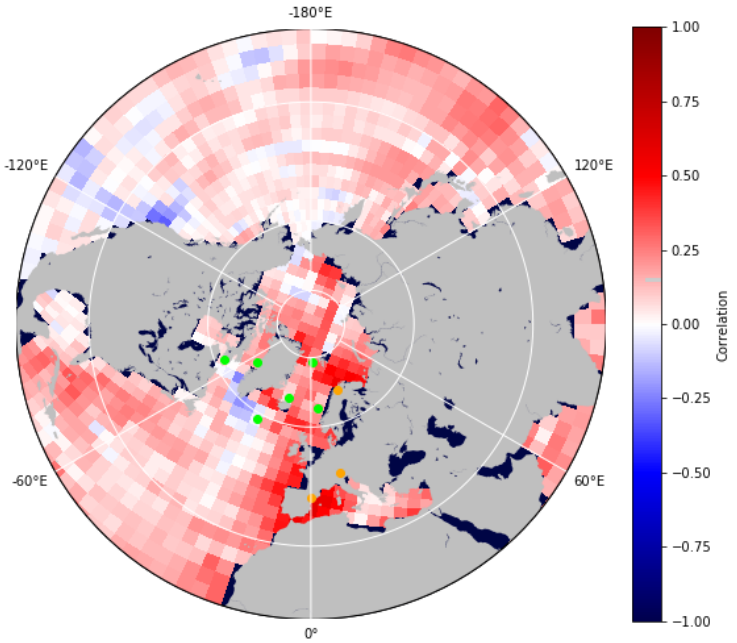

Figure 24: Illustration of yearly Correlation Coefficients on SST, cumulative

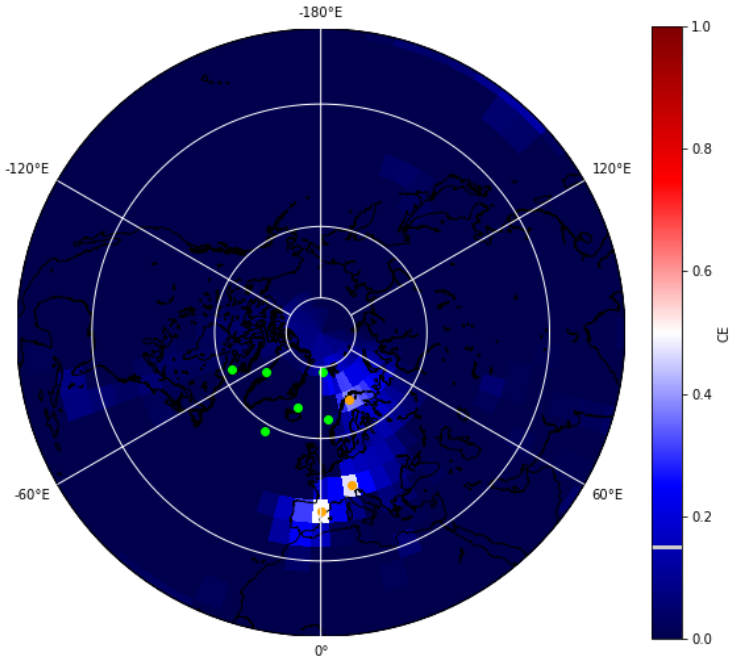

Figure 25: Illustration of yearly CE on TS, cumulative

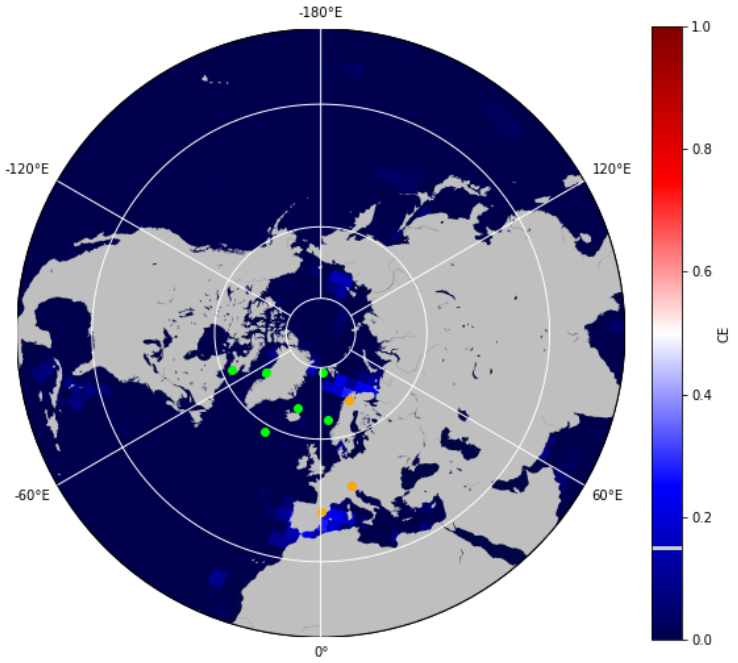

Figure 26: Illustration of yearly CE on SST, cumulative

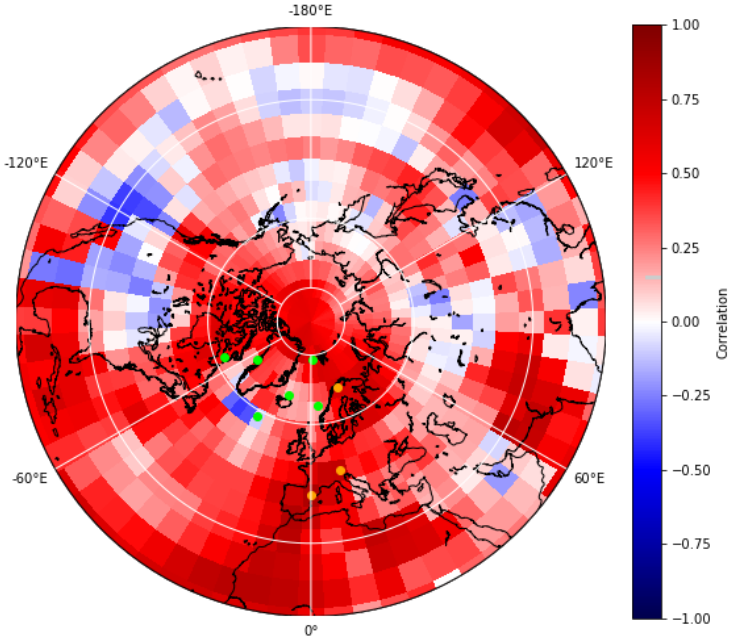

Figure 27: Illustration of decadal Correlation Coefficients on TS, cumulative

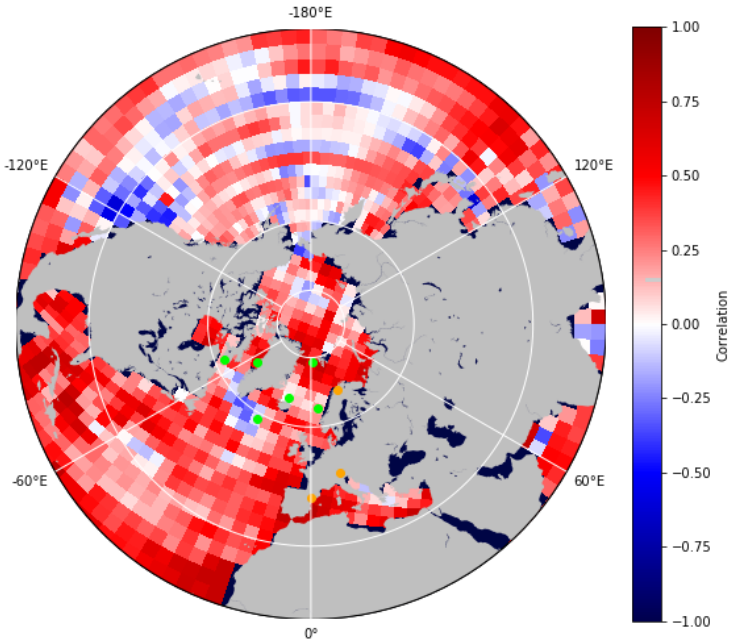

Figure 28: Illustration of decadal Correlation Coefficients on SST, cumulative

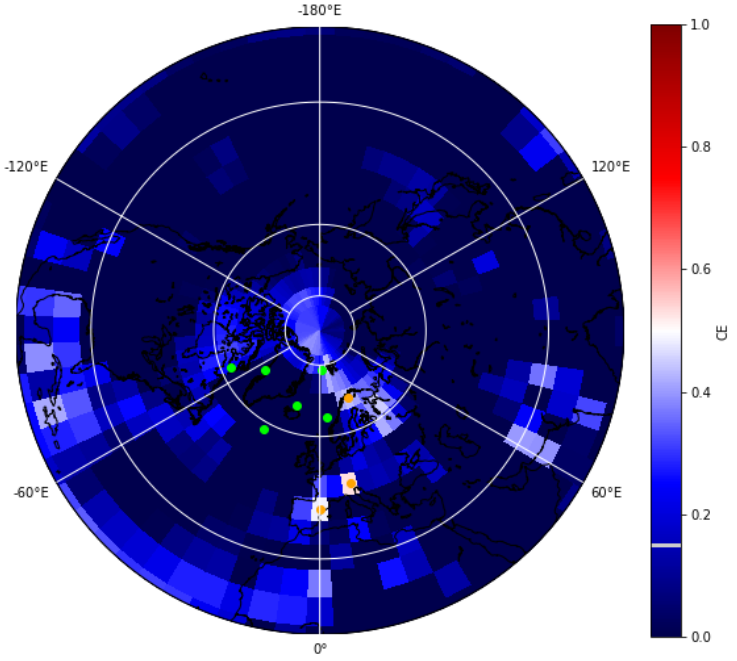

Figure 29: Illustration of decadal CE on TS, cumulative

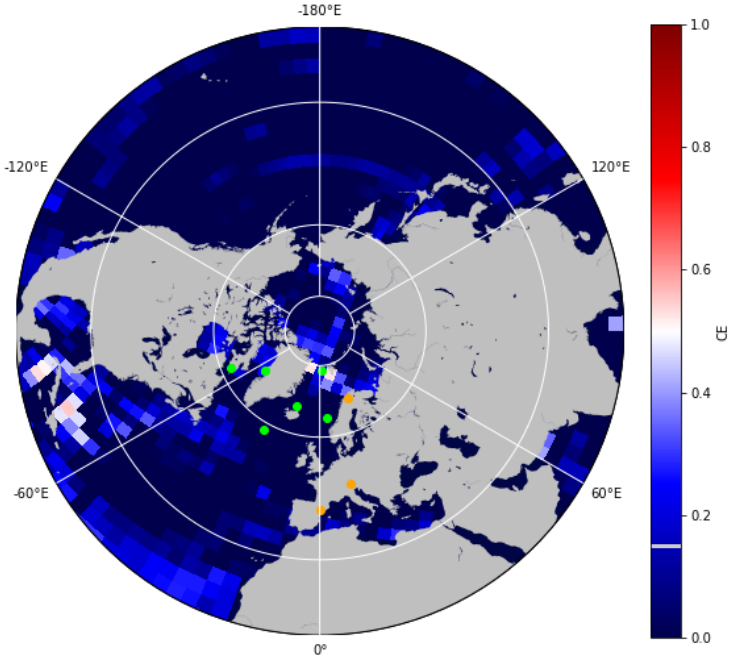

Figure 30: Illustration of decadal CE on SST, cumulative