INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Data Assimilation for Numerical Weather Prediction

by

N. Brahma Reddy

A thesis submitted in fulfillment for the
Masters Degree

Under Guidance of
Prof. Sourangshu Bhattacharya
Department of Computer Science

April 16, 2015

Data Assimilation for Numerical Weather Prediction

Thesis submitted by

N. Brahma Reddy

13CS60R09

M.Tech CSE

Approved by

Prof. Sourangshu Bhattacharya

(Supervisor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Declaration of Authorship

I, N.Brahma Reddy, declare that this thesis titled, 'Data Assimilation for Numerical Weather Prediction' and the work presented in it are my own. I confirm that:

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

Signed: _____

Date: _____

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Certificate

This is to certify that the project entitled **"Data Assimilation for Numerical Weather Prediction"** is a bonafide record of the work carried out by Mr. N.Brahma Reddy (Roll No.13CS60R09) under my supervision and guidance for the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science & Engineering during the academic session 2013-2015 in the Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur.

The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Prof. Sourangshu Bhattacharya

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Kharagpur, India 721302

April 16, 2015

“”When I’m working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong.“”

,R. Buckminster Fuller

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Abstract

Prof. Sourangshu Bhattacharya

Department of Computer Science

Masters Degree

by N. Brahma Reddy

During the last 20 years data assimilation has gradually reached a mature center stage position at Numerical Weather Prediction centers. In this report I provided a short survey on Numerical Weather Prediction (NWP) and models that are used in NWP, and Data Assimilation Techniques that are useful in NWP, different types of assimilation techniques. Main aim of this project is to assimilate the daily data, by considering first guess and few observations at sparse locations. Besides several techniques Complete Implementation of Optimal Interpolation Technique, Different type of data required for prediction. Finally included results i.e. predicted sea temperature at different depths,errors in each prediction, and RMS error per each day Including all depths....

Acknowledgements

This thesis is the result of the project work performed under the guidance of Prof. Sourangshu Bhattacharya at the Department of Computer Science and Engineering of the Indian Institute of Technology, Kharagpur. I am deeply grateful to my supervisor for having given me the opportunity of working under him and guiding me seamlessly for the whole period. He gave me exposure to all the related research going on in this field, which helped me enormously. Without the help,encouragement and patient support I received from my guide, this report would never have materialized.

I am very much thankful to Prof. Arun Chakraborty, Head of the Department of Center for Oceans, Rivers, Atmosphere and Land Sciences, IIT Kharagpur for providing guidance regarding weather data and verification of results, which helped me very much improve on this thesis work.

I am very much thankful to Prof. Rajib Mall, Head, Department of Computer Science & Engineering Department, IIT Kharagpur for providing necessary facilities during the research work.

I am thankful to Faculty Advisor Prof. Sudeshna Sarkar and all the faculty members of the department for their valuable suggestions, which helped me improve on this research work.

I express my sincere gratitude to Mr.Tarumay Ghoshal and Mr.Chandan Misra for their valuable guidance, continuous support and encouragement, throughout the course of this research work.. . .

Contents

List of Figures

Abbreviations

NWP	**N**uemerical **W**eather **P**redictoin
OI	**O**ptimal **I**nterpolation
RMS	**R**oot **M**ean **S**quare

Symbols

N	Number of Grid Points	
P	Number of Observational Points	
R	Radius of Influence	
W	Weight Matrix	$[N \times P]$
H	Interpolation Matrix	$[P \times N]$
X_b	Background Vector	$[N \times 1]$
X_t	True Vector	$[N \times 1]$
Y_o	Observational Vector	$[P \times 1]$
X_a	Analysis Vector	$[N \times 1]$
e_a, e_b, e_o	error in corresponding Vectors	

Chapter 1

Introduction

1.1 Motivation

- For more accuracy in prediction of cyclones, and ocean circulation patterns high quality daily temperature, salinity data required.
- For improving the cyclone track predication we need improvement in existing data with insertions of ground truth observations.
- Daily observations are available only at sparse locations.
- Efficient data assimilation algorithm is required to combine these in the existing data set.
- In this Project I implemented such algorithm, Optimal interpolation, by using parallel processing.

1.2 Introduction

Numerical weather prediction uses mathematical models of the atmosphere and oceans to predict the weather based on current weather conditions. Though first attempted in the 1920s, it was not until the advent of computer simulation in the 1950s that numerical weather predictions produced realistic results. A number of global and regional forecast models are run in different countries worldwide, using current weather observations relayed from radiosonde or weather satellites as inputs to the models. Operational NWP centers produce initial conditions through a statistical combination of observations and short-range forecasts. This approach has become known as "data assimilation".

Chapter 2

Numerical Weather Prediction

NWP is an initial value problem. Provided an estimate of the atmospheric state, in terms of the variables of the NWP model, the model simulates the atmospheric state at later times. It also calculates precipitation and other important properties used by weather forecasters in the production of the public weather forecasts.

NWP is focused on taking current observations of weather and processing these data with computer models to forecast the future state of weather. Knowing the current state of the weather is just as important as the numerical computer models processing the data.

2.1 Model

Tool for simulating or predicting the behavior of a dynamical system such as the atmosphere.Types of models include:

Heuristic : Rule of thumb based on experience or common sense

Empirical : Prediction based on past behavior

Conceptual : Framework for understanding physical processes based on Physical reasoning

Analytic : Exact solution to "simplified" equations that describe the Dynamical system

Numerical : integration of governing equations by numerical methods Subject to specified initial and boundary conditions

2.2 Observation Data

Like medicine, a Weather Prediction needs a diagnosis based on observation. In the case of weather this involves a complex system of measurements of the atmosphere from terrestrial and space based systems.

2.2.1 Terrestrial based observing

Terrestrial based observing Weather observations have for many years been made from networks of stations over land and from ships on passage.

The conventional method of measuring wind, temperature and humidity above ground level is the Radio-sonde. A radiosonde is a small weather station coupled with a radio transmitter. The radiosonde is attached to a helium or hydrogen-filled balloon, generally called a weather balloon, and the balloon lifts the radiosonde to altitudes exceeding 115,000 feet.

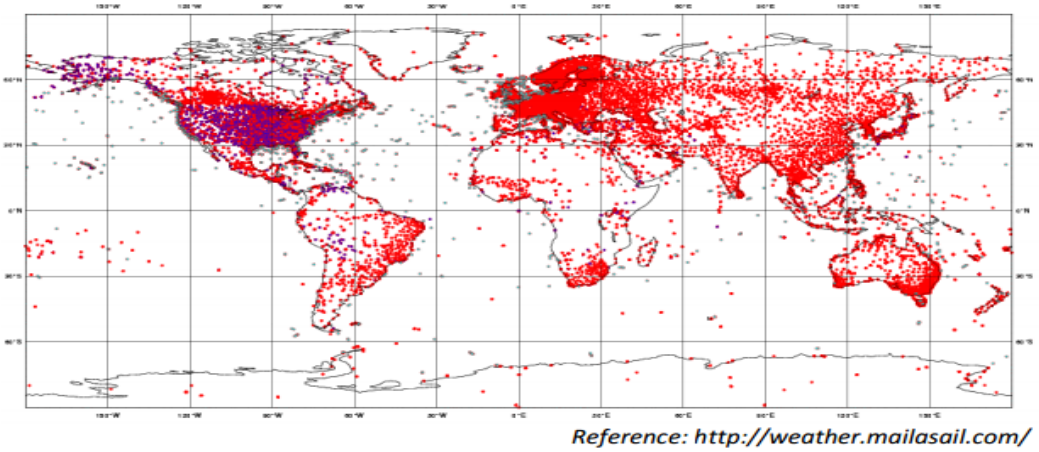


FIGURE 2.1: Scattered Observational points world wide

2.2.2 Space Based systems (Sensors)

Sensors can be passive or active. Passive sensors "look" at the earth and atmosphere using visible, infrared and microwave frequencies. Active sensors transmit a signal and receive the return; these act lie radar.

2.2.3 Space Based systems (Orbits)

Orbits can be geostationary or low earth, usually; near polar orbit at heights of 400 – 800 m Geostationary satellites are good at looking at cloud or water vapor and measuring

movements to deduce winds at cloud levels. At 35, 000 km they are too high to be able to produce good measurements otherwise.

Low earth orbiters provide most of the temperature and humidity data using passive sensors.They use active sensors to measure winds at sea level. This is done by measuring the scattering of a radar beam using a "Scatterometer."

2.3 Components of NWP Model

- Governing Equations
  - $F = ma$, conservation of mass, moisture, and thermodynamic eqn. gas law
- Numerical procedures
  - Approximations used to estimate each term (especially important for advection terms)
  - Approximations used to integrate model forward in time
  - Boundary conditions
- Approximations of physical processes (parametrization)
- Initial conditions
  - Observing systems, objective analysis, initialization, and data assimilation

Chapter 3

Data Assimilation

Purpose of data assimilation: using all the available information, to determine as accurately as possible the state of the atmospheric (or oceanic) flow The need for an automatic "objective analysis" became quickly apparent (Charney, 1951), and interpolation methods fitting observations to a regular grid were soon developed. Panofsky (1949) developed the first objective analysis algorithm based on two-dimensional polynomial interpolation, a procedure that can be considered "global" since the same function is used to fit all the observations.

Gilchrist and Cressman (1954) developed a "local polynomial" interpolation scheme for the geopotential height.

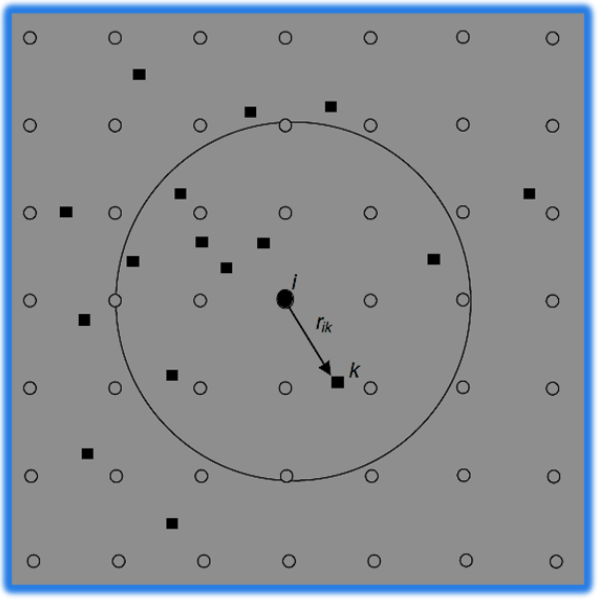


FIGURE 3.1: Schematic of grid points,and observations.

here the circle represents the grid points, the squares represents the scattered observations, and and a radius of influence around a grid point i marked with a black circle.

the grid-point analysis is a combination of the forecast at the grid point (first guess)and the observational increments (observation minus first guess) computed at the observational points k.

3.1 Analysis Cycle

The analysis cycle is an intermittent data assimilation system that continues to be used in most global operational systems, which typically use a $6h$ cycle performed four times a day.

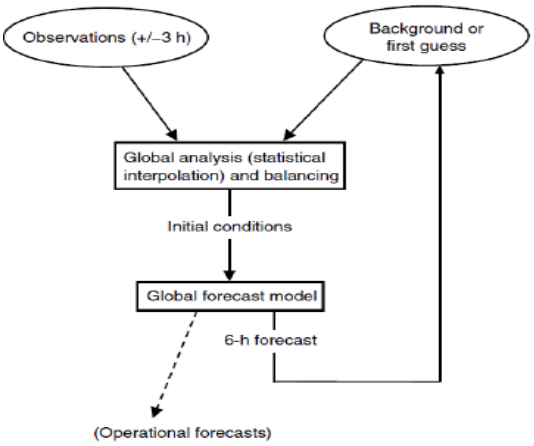


FIGURE 3.2: Global Analysis Cycle

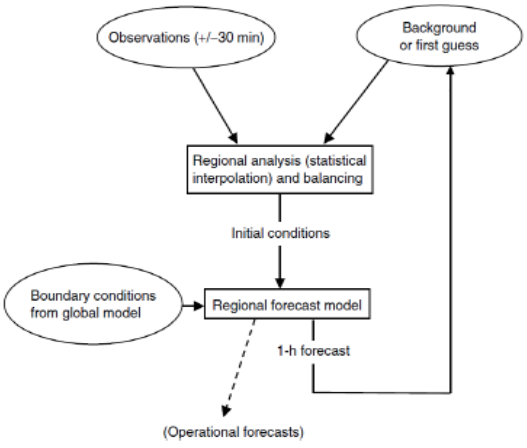


FIGURE 3.3: Regional Analysis Cycle

- The first picture shows, Typical global 6-h analysis cycle performed at 00, 06, 12, and 18 UTC. The observations should be valid for the same time as the first guess.
- Second one shows, Typical regional analysis cycle. The main difference with the global cycle is that boundary conditions coming from global forecasts are an additional requirement for the regional forecasts.

- Observations
  - we can get observational values as mentioned in Chapter 2 Successive Correction Method (SCM)
  -

- Background Values
  - First guess or prior information, The Background is outcome of stabilized models, in these models considers some initial conditions, and integrating governing equations on these conditions. Objective Analysis(OA) is example of such Model. The output of these models is in gridded form.

3.2 Data Assimilation Methods

A large number of data assimilation methods exist for Oceanography . Which method is adopted depends mainly on the NWP model in question (type, area) and the available resources for data assimilation. It is a significant fraction of the total NWP computing time which is spent on data assimilation, wherefore time constraints play a dominating role when selecting data assimilation method. Mainly for this reason adaptive statistical methods are not yet used operationally. The methods can be divided into classes:

- Empirical methods
  - Successive Correction Method (SCM)
  - Nudging
- Constant statistical methods
  - Optimal interpolation (OI)

3.2.1 Empirical methods

3.2.1.1 Successive Correction Method

In the successive corrections method, the field variables are modified by the observations in an iterative manner. A pass is made through every grid point, updating the variable at each grid point based on first guess field and the observations surrounding that grid point.

$$f_i^{m+1} = \frac{\sum_{p=1}^{P} w_{ip}(O_p - f_{pm})}{\sum_{p=1}^{P} w_{ip} + \epsilon^2}$$

where f_i^m is the value of the variable (e.g., T, q, u, etc.) at the i^{th} grid point at the m^{th} iteration, O_k is the k^{th} observation surrounding the grid point, w_{ik}^m is a weighting function which depends on how far the observation is from the grid point, and ϵ^2 is an estimate of the ratio of the observation error to the first guess field error (if the observations were perfect then $\epsilon^2 = 0$).

3.2.1.2 Nudging

The Nudging[3] method (also known as Newtonian relaxation and dynamic initialization) adds a nudging term to the prognostic equations for the field variables. The simulation is initialized with the first guess field, and the equations are integrated forward. The nudging term forces the integration towards the observations. This method also balances the initial conditions, since as they are integrated forward the fields will adjust geostrophically and hydrostatically. The prognostic equations would look something like

$$\frac{da}{dt} = F(a,t) + G(t)\sum_{i}^{N} w_i(a_i - a)$$

where the term on the left hand side of the equation is the model tendency, F(a,t) is the model forcing, and the final term in the nudging term. G(t) is the nudging coefficient, wi is an analysis weight, ai is an observed value, and a is the interpolated model value.

3.3 Optimal Interpolation

Optimal Interpolation consists of taking into account (assimilate) the new information that the observational data provide in order to advance in time the "background" state (also called first guess or prior information) that the weather forecasting numerical code has predicted. The increment is obtained by taking the difference or innovation between the observational data and the observational operator.

The new state or analysis is then the result of the assimilation/forecast procedure. More specifically, let x_b be the background vector state characterizing the current state of the model,$H(x_b)$ the observational operator and y_o the observational data to be assimilated in the model,then one can show that the analysis xa is

$$x_a = x_b + W(y_o - H(x_b))$$

with W the weights determined from the estimated statistical error covariance's of the forecast and the observations (Kalnay 2003).

The assimilation methods consist of predicting the evolution of the errors and of course of minimizing it, i.e. keeping it under control as much as possible given the very chaotic nature of the Earth's atmosphere

Chapter 4

Implementation of Optimal Interpolation

4.1 ALGORITHM

4.1.1 INPUTS:

1. **Back Ground Vector** x_b **:** Background Values at Grid points, N-Dimensional Vector.
   (a) Output of Object Analysis Model.
   (b) It's a matrix of size $[longitude \times latitude \times depth]$, we need to convert to N-Dimensional Vector.
2. **True Vector** x_t **:** True Values at Grid points, N-Dimensional Vector
   (a) Data Source from ASIA-PACIFIC DATA-RESEARCH CENTER (APDRC).
   (b) It's a matrix of size $[longitude \times latitude \times depth]$, we need to convert to N-Dimensional Vector.
3. **Observational Vector** y_o **:** Observational Values at scattered locations and for a single location several depths, P-Dimensional Vector.
   (a) Data Source from WORLD OCEAN DATABASE
   (b) It's a matrix of size e $[No.of locations \times No.of Depths]$, we need to convert to P-Dimensional Vector.

4.1.2 ALGORITHM:

1. **Calculate the interpolation matrix :**

   (a) The Interpolation matrix is of size of $[P \times N]$ or $[No.ofObservations \times No.ofGridpoints]$

   (b) Interpolation matrix $H = \begin{bmatrix} h_{11} & h_{12} & ... & h_{1n} \\ h_{21} & h_{22} & ... & h_{2n} \\ . & . & ... & . \\ h_{p1} & h_{p2} & ... & h_{pn} \end{bmatrix}$

   (c) Computing Interpolation matrix:

      i. Every element of interpolation matrix indicates weighted average of distance.

      ii. For a single observation point we will compute distance to every grid point, if the distance is less than or equal to the "radius of influence" assign $h_{ij} = dist(i,j)$ distance between i^{th} observation point and j^{th} grid point. And maintain a variable called $totalweight$ initially 0, and add distance to $totalweight = totalweight + dist(i,j)$.

      iii. Else if distance greater than the "radius of influence" just $h_{ij} = 0$ and don't include $dist(i,j)$ to $totalweight$

      iv. At the end i.e. after completion of one complete row divide all values of the row by $totalweight$, and make $totalweight = 0$ for the next rows. Hence we can write h_{ij} like

$$h_{ij} = \begin{cases} \frac{dist(i,j)}{\sum_{k \in allpointswithinR} dist(i,k)}, & \text{if } dist(i,j) \leq R \\ 0, & \text{otherwise} \end{cases}$$

2. By using the calculated Interpolation Matrix(H) compute two new vectors:

   (a) $y_t = H \times x_t$ Interpolate given True values to Observational Co-ordinates.

   (b) $y_b = H \times x_b$ Interpolate given Background values to Observational Co-ordinates.

3. Calculate the error in the Background vector and Observational vector:

   (a) $e_b = x_b - x_t$ Error in the given background vectors (error always calculated by comparing with true values).

   (b) $e_o = y_o - y_t$ Error in the given observational vector.

4. Compute Background error co-variance matrix, Observational error co-variance matrix:

   (a) $B = E[e_b \times e_b^\intercal]$, a $N \times N$ matrix. Here for Expectation we are just considering Average i.e. $B = \frac{[e_b \times e_b^\intercal]}{N}$.

   (b) $R = E[e_o \times e_o^\intercal]$, a $P \times P$ matrix. Here for Expectation we are just considering Average i.e. $R = \frac{[e_o \times e_o^\intercal]}{P}$.

5. Compute Weight(W) Matrix:

   (a) $W = B \times H^\intercal \times (R + H \times B \times H')^{-1}$, a $N \times P$ matrix.

4.1.3 OUTPUT:

1. The Analysis vector can be computed by using weight matrix

   (a) $x_a = x_b + W \times [y_0 - y_b]$, x_a is is a N-Dimensional vector, same as Background Vector

2. Error in analysis vector and co-variance matrix

   (a) Same as error in Background vector we can compute error in the analysis vector like $e_a = x_a - x_t$.

   (b) the co-variance matrix can be calculated as $A = B - W \times H \times B$, where I is Identity matrix.

Chapter 5

Optimizations and Parallel Processing

5.1 Optimizations

5.1.1 Computing Interpolation Matrix

- **Brute force :** While computing interpolation matrix[$P \times N$] for every point in observation point we need to find the points which are within the Radius of Influence.
- Find distance to every grid point from a single observation point and assign value to points having less distance than R, since there are P observations it will take O(PN) time.
- **KD-trees :** Computation of interpolation matrix can be done by using Kd-trees efficiently. construct the Kd-tree for all grid points, and then do range search for every observation point. Constructing and Range search take only O($N \log N + P \log N$).

5.1.2 Computing Weight Matrix

- Computation of weight matrix $W = B \times H^{\intercal} \times (R + H \times B \times H')^{-1}$ will requires $2np(n+p) + p^3$ number of multiplication
- Consider the matrix $H \times e_b$ as A, and W matrix will becomes

$$
\begin{aligned}
W &= B \times H^{\intercal} \times (R + H \times B \times H')^{-1} \\
&= e_b \times e_b^{\intercal} \times H^{\intercal} \times (R + H \times e_b \times e_b^{\intercal} \times H')^{-1} \\
&= e_b \times A^{\intercal} \times \ (R \ + \ A \times \ A^{\intercal})^{-1}
\end{aligned}
$$

- For the same weight matrix, now it requires only $p(n+p)+p^3$ multiplications.

5.1.3 Filling the missing values

the availability of the data is one of the main problem in numerical weather prediction.many times we can't get data at every gird point. so the prediction accuracy will reduce, to avoid this problem we can fill those missing values (NaN) by any analysis model. here we are using through Branes objective analysis model.

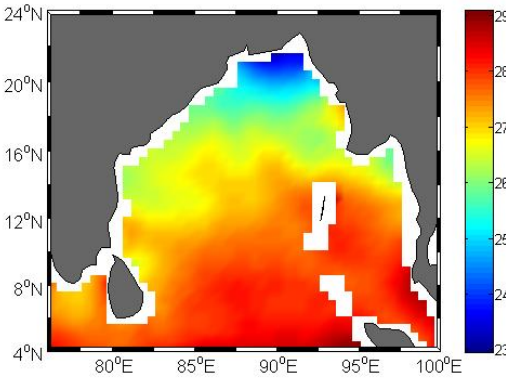


FIGURE 5.1: Before filling the missing(NaN) values

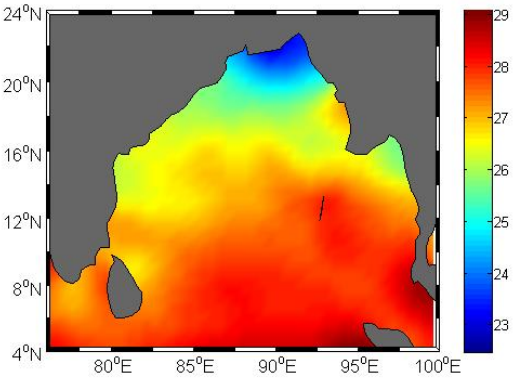


FIGURE 5.2: After filling the missing(NaN) values

5.2 Parallel Processing

5.2.1 Problems with linear processing

Since the data assimilation is involves more time and space consuming operations like large number of matrix multiplications and inverse of matrix. if the data size more it becomes more tedious to perform these operations. and computing inverse in single system with tools like matlab not possible. consider data like $P = 10k$ and $N = 20k$ computing the inverse will consume more memory which causes system crash.and same problems will occur for multiplication of huge matrices.

5.2.2 Parallel Processing

To avoid the problems of linear programming we can use parallel programming. mathematical operations like finding inverse, multiplying huge matrices can be done in big-data platform like *spark* or *hadoop*.

in parallel processing we decompose the single huge problem into several small small tasks and execute them parallel in different systems and combine for the final answer. since we are decomposing the total task into tiny tasks and each task requires much less memory compare to the initial problem so we don't need much memory.but if we have less number of resources i.e. systems it may take more time since there may be large number of tiny tasks to complete. but for more memory systems we can divide less number of tasks so we can compute in less time successfully.Complete thesis work done in *apache-spark* platform. complete source code can be found in `https://github.com/krishna0545/OptimalInterpolation`.

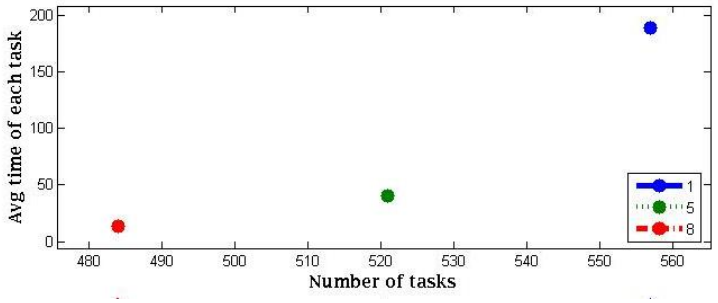


FIGURE 5.3: Performance,after addition of more clients

the plot shows the observations of finding inverse of a $[20k \times 20k]$ matrix.Shows the improvement in performance by adding more number of clients.the plot is drawn for *Number of small tasks* from the division of original problem by apache-spark Vs *average time for each task*. the red dot represents cluster with 8 clients.

the green dot shows a cluster with 5 clients, the blue dot shows cluster with only one client. and each client contributes 512Mb memory for execution of total task.from picture we can observe that more the memory lesser the time of computation.The figure 5.4 shows the plot between Number of clients and Total time of computation.

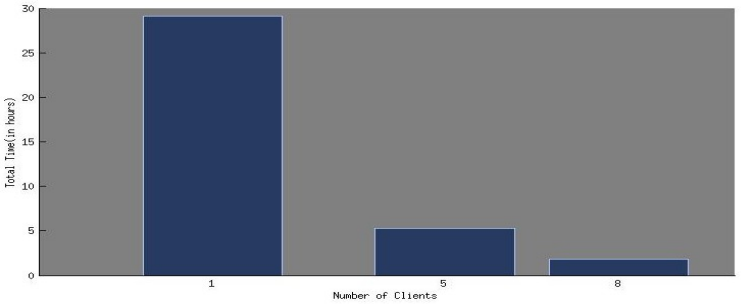


FIGURE 5.4: Performance,after addition of more clients

5.2.2.1 Finding Inverse of Matrix

During the computation of the weight matrix we need to find the inverse of matrix.

$$W = e_b \times A^\intercal \times \ (R \ + \ A \times \ A^\intercal)^{-1}$$

finding the inverse in tools like matlab can be done by using inbuilt packages. but the same operation in parallel processing becomes challenging task since there is no proper algorithm to find the inverse of the matrix.

For finding the Inverse we have used SVD decomposition in this thesis.finding the inverse mathematical way can be explain as,
Solving x for the linear equation:

$$\begin{aligned} Ax &= b \\ x &= bA^{-1} \end{aligned}$$

Since SVD decomposition of the A=UsV

$$\begin{aligned} &= b(UsV)^{-1} \\ &= b \times (V)^{-1} \times s^{-1} \times U^{-1} \end{aligned}$$

Since U,V are Orthogonal matrices i.e. $U \times U^\intercal = I \ and \ V \times V^\intercal = I$

$$= b \times V^\intercal \times s^{-1} \times U^\intercal$$

Since s is a diagonal matrix its inverse is matrix with inverse diagonal values i.e

$$(diag[a_1, a_2, a_3..a_n])^{-1} = diag[\frac{1}{a_1}, \frac{1}{a_2}, \frac{1}{a_3}...\frac{1}{a_n}]$$

So x value becomes

$$x = b \times V^\intercal \times S_{inv} \times U^\intercal$$

now finding the inverse is just multiplication of matrices.

Chapter 6

Results and Comparison

6.1 Comparison

6.1.1 Both serial and parallel systems produce same output

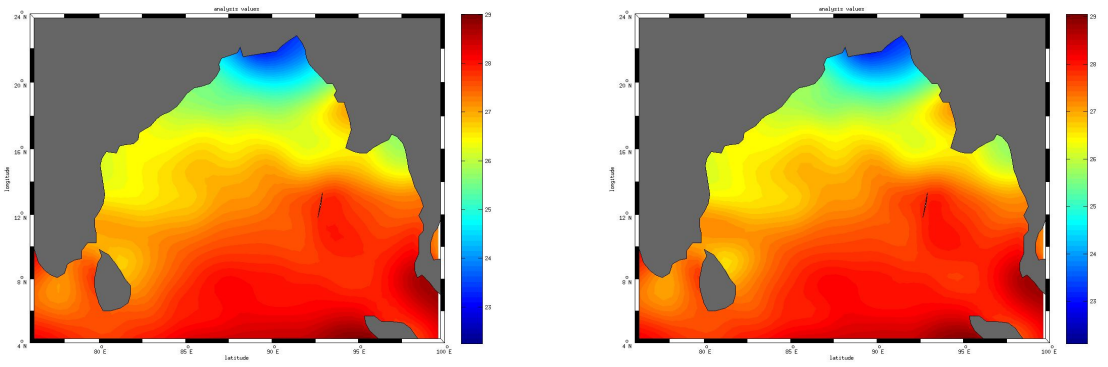


FIGURE 6.1: the analysis values of at the depth $0m$. *left:*matlab,*right:*spark

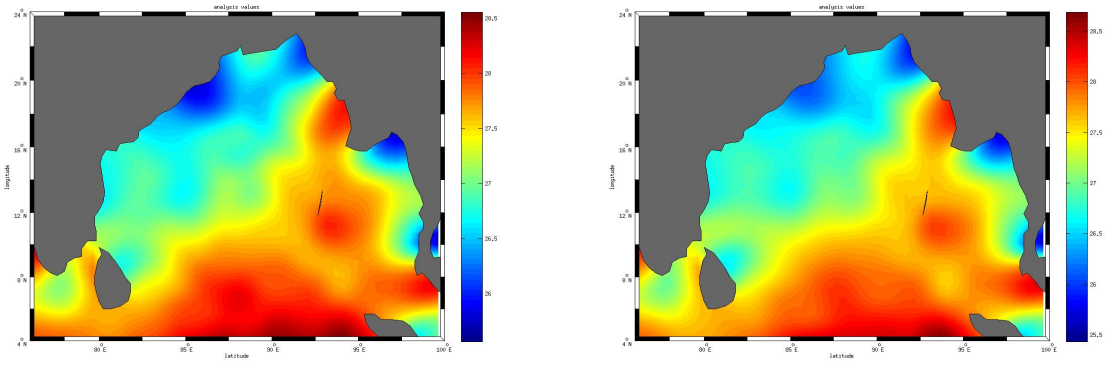


FIGURE 6.2: the analysis values of at the depth $40m$. *left:*matlab,*right:*spark

the first row figures shows the analysis values on Tuesday 1st January, 2013 at the depth $0m$, left one is output of matlab tool and right one shows output through parallel execution($spark$), similarly the second row figures shows the analysis values on same date at the depth $40m$, left one is results through matlab tool and right one shows output through parallel execution($spark$).

from the above pictures we can observe that the both ways of data assimilation will produce same results.

6.2 Results

6.2.1 Error in Analysis

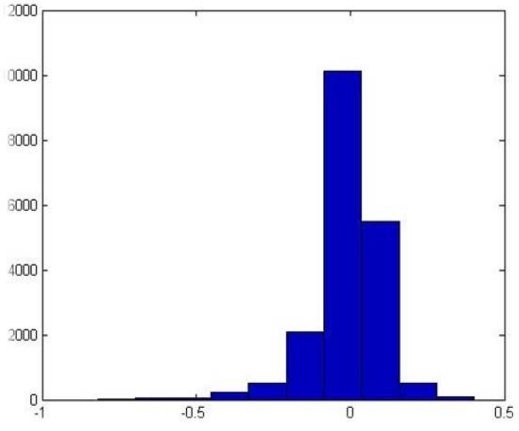


FIGURE 6.3: Error in Analysis Values for the data on $Jan\ 1^{st}$

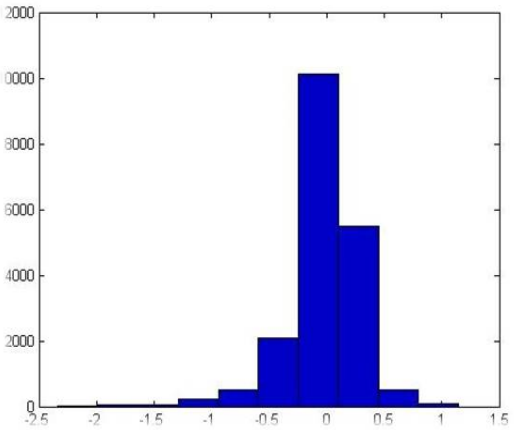
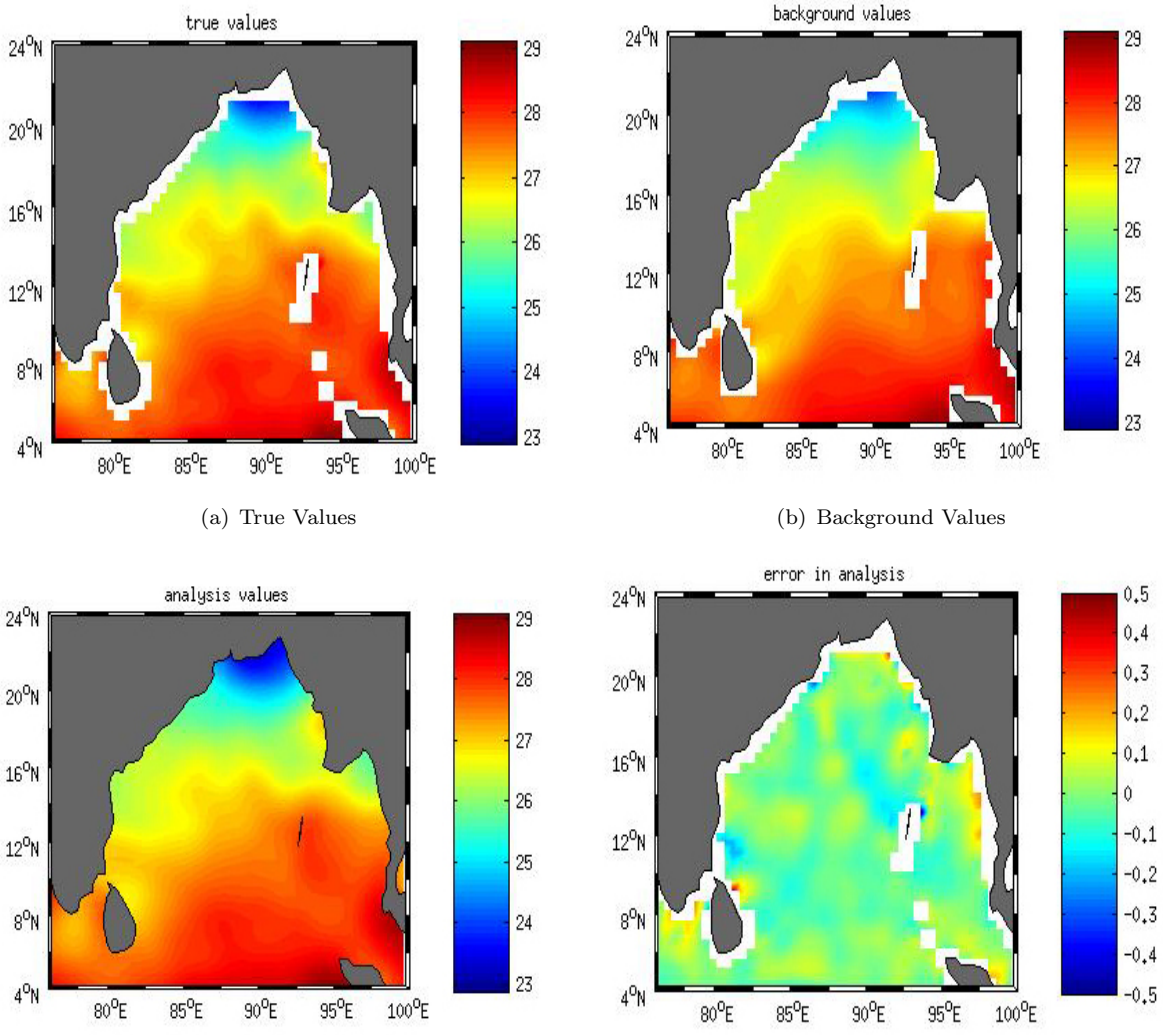


FIGURE 6.4: Error in Analysis Values for the data on $Jan\ 2^{nd}$

the first row figures shows the error in analysis values on $jan\ 1^{st}$ 2013, the second figures shows the error in analysis values $jan\ 2^{nd}$ 2013. from the figures we can say error is in skew-distribution having the peak around 0 on both days.

6.2.2 Data Assimilation of temperature, Jan 1st 2013

6.2.2.1 Depth 0 meters



(a) True Values

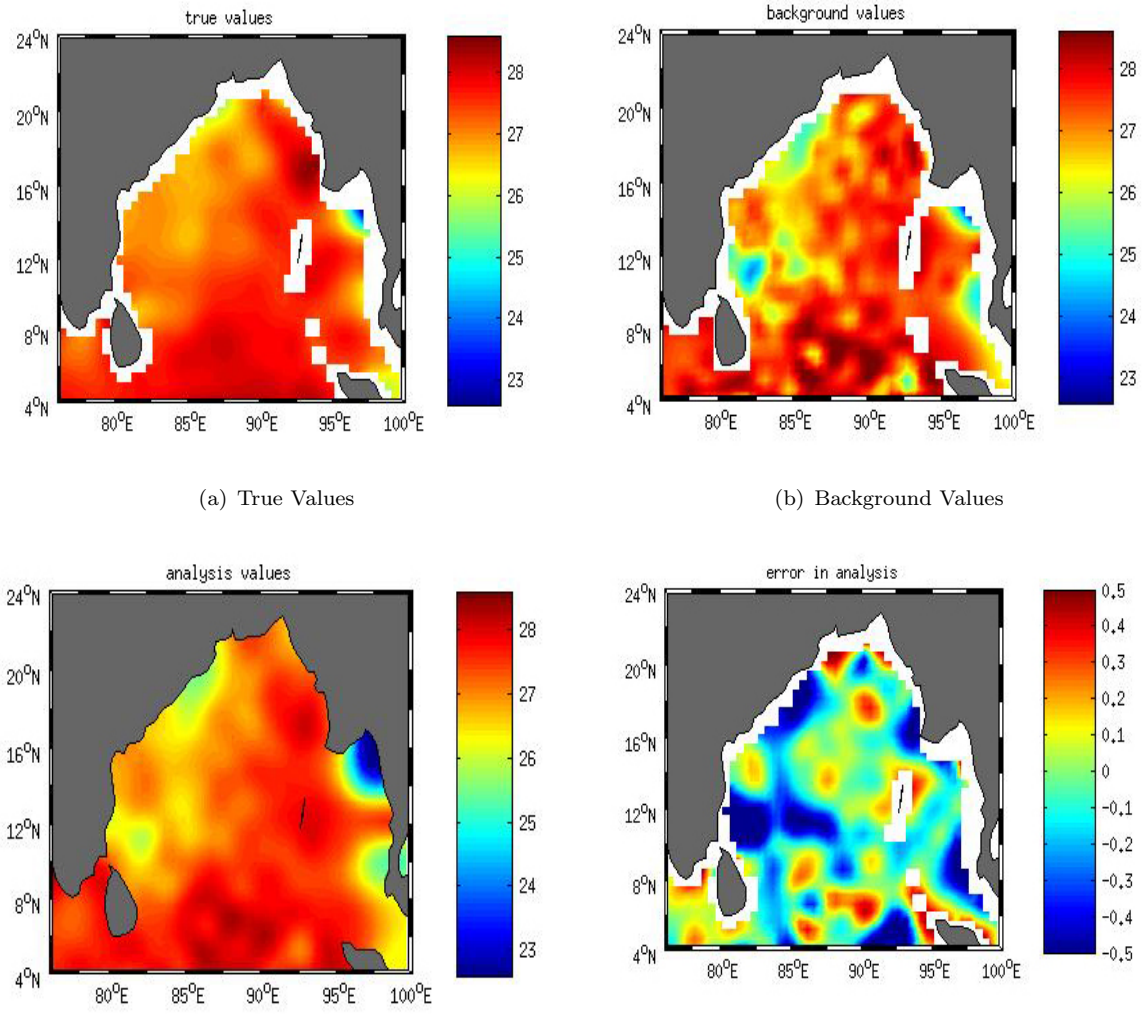
(b) Background Values

(c) Analysis Values

(d) Error in Analysis

FIGURE 6.5: Data Assimilation of temperature at depth 0m

6.2.2.2 Depth 40 meters



(a) True Values

(b) Background Values

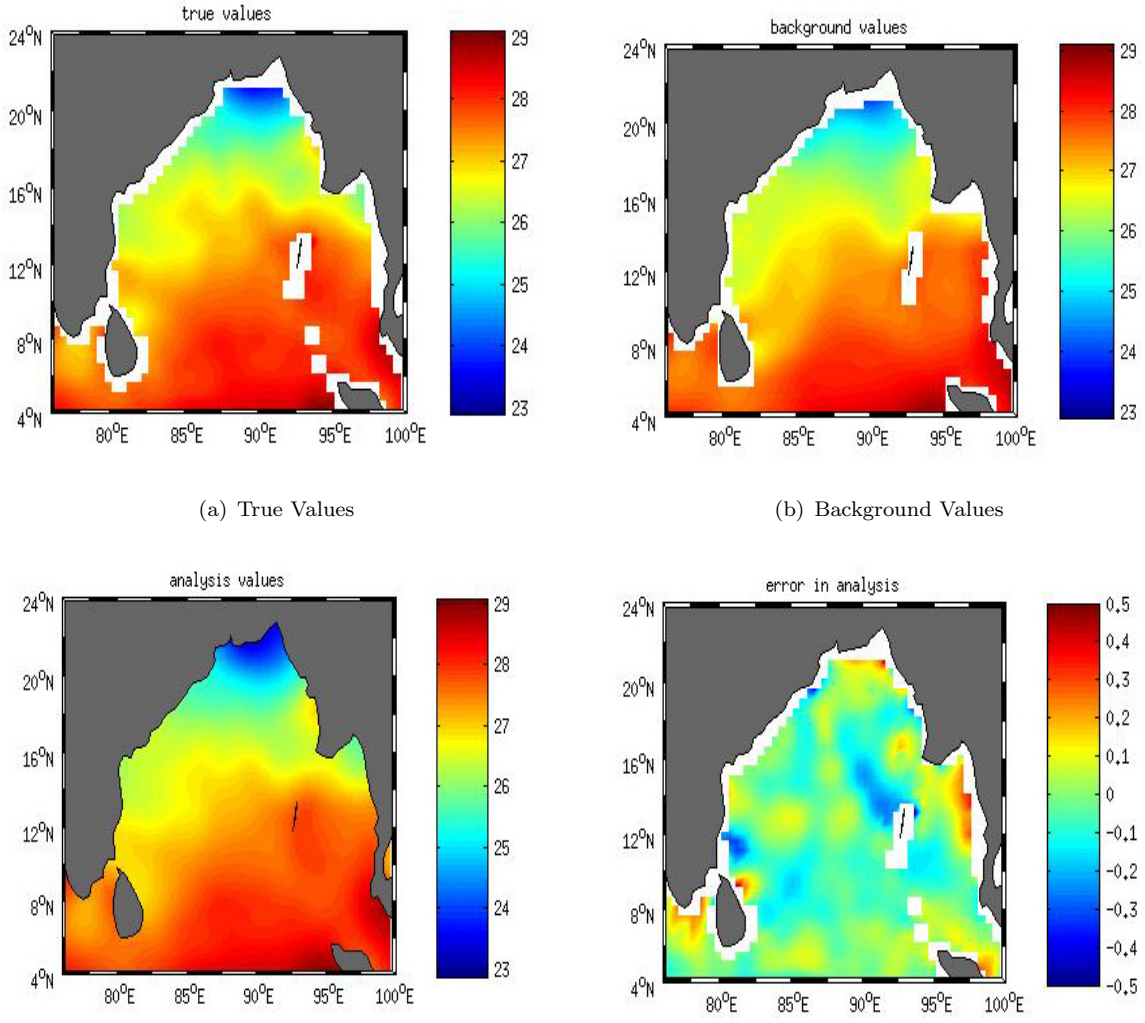
(c) Analysis Values

(d) Error in Analysis

FIGURE 6.6: Data Assimilation of temperature at depth 40m

6.2.3 Data Assimilation of temperature, Jan 10th 2013

6.2.3.1 Depth 0 meters



(a) True Values

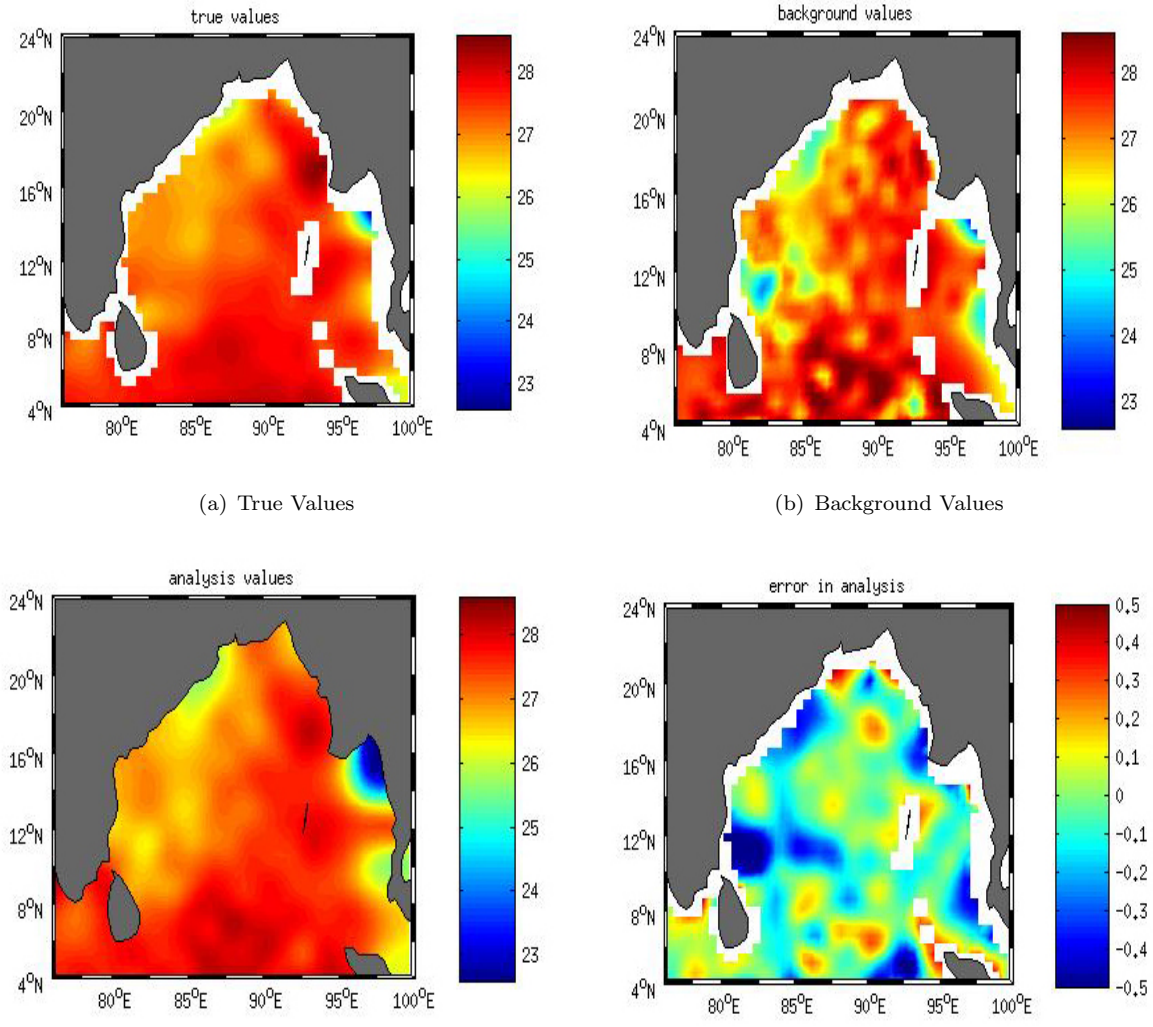
(b) Background Values

(c) Analysis Values

(d) Error in Analysis

FIGURE 6.7: Data Assimilation of temperature at depth 0m

6.2.3.2 Depth 40 meters



(a) True Values

(b) Background Values

(c) Analysis Values

(d) Error in Analysis

FIGURE 6.8: Data Assimilation of temperature at depth 40m

the last 4 pages includes the data assimilation results on several dates and several depths, since at lower depths there may not be much availability of temperature values, we can observe that the more the depth there is increase in error and error is not fallowing any pattern.

6.3 RMS error for assimilation of temperature

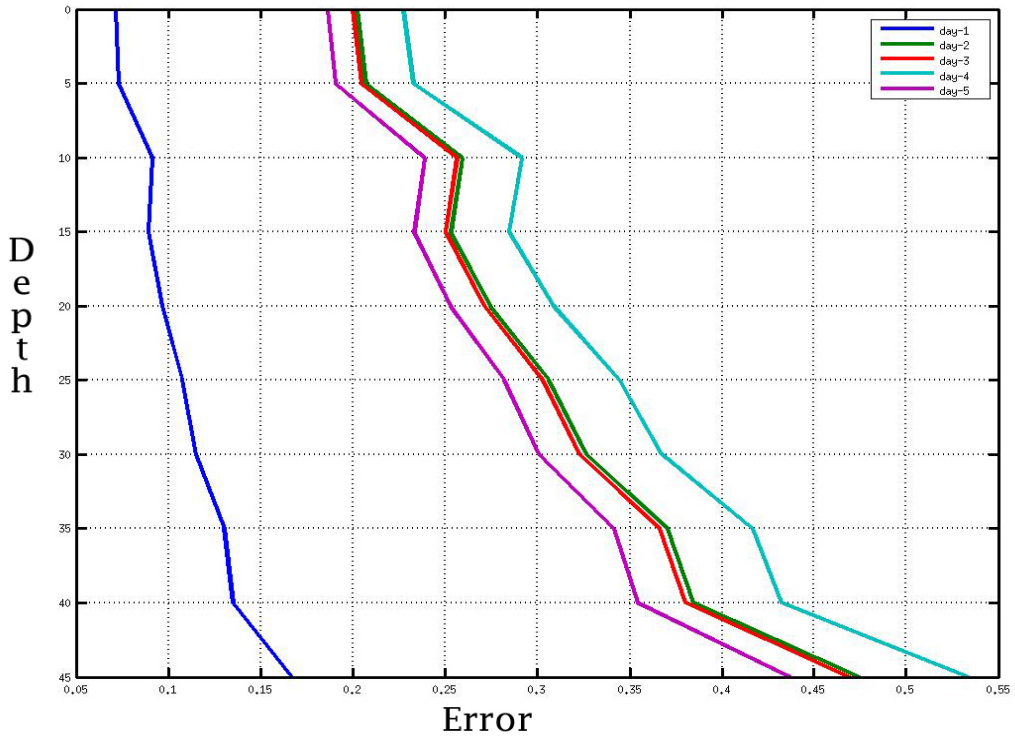


FIGURE 6.9: RMS error for first 5days of January

Each color represents a single day rms error, the plot is drawn *Error Vs Depth.* and it is clear that with increasing of depth the error is increasing slightly.

6.4 Time of Computation

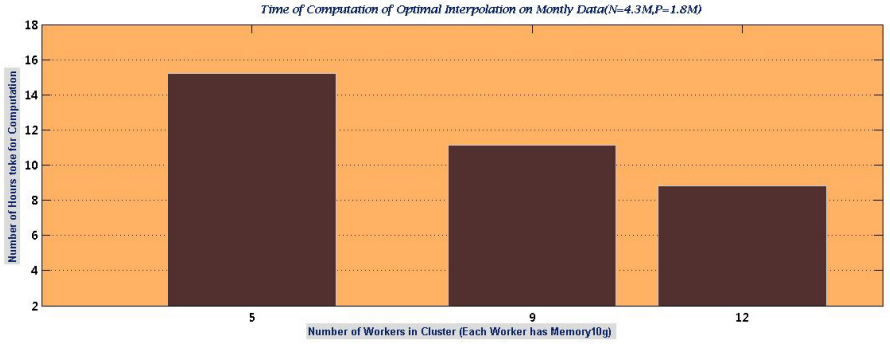


FIGURE 6.10: total time of computation of Optimal Interpolation on different cluster sizes

The figure shows the time taken for computation of Optimal Interpolation algorithm on different sizes of clusters.in the picture X-axis represents the size of cluster i.e. number of workers attached to sever, here each worker has memory of 10, and the master has 20g and total cluster works with 6 cores.the Y-axis represents the number

of hours it took to generate the results. the Interpolation algorithm is for the monthly data where the number of gridded points are $N \approx 4.3 \times 10^6$ and number of observations $P \approx 1.8 \times 10^6$. the first bar shows time for cluster size 5 which is 15.23Hrs the second bar indicates for cluster size of 9 which is 11.15Hr and the final bar represents time for cluster size of 12 workers which is 8.80Hrs.

Chapter 7

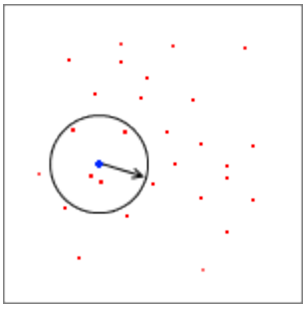
Conclusion

The data assimilation is normally used to increase the accuracy in prediction.the data assimilation can be done by using several interpolation algorithms in this thesis we have used linear Optimal Interpolation in this we uses linear equation to convert scattered observations.Since we may have large number of data points and observational points throughout month/year it may not be feasible or we cant compute some operations on single machine.even though for smaller data sets the single process computing tools like matlab gives better results comparatively parallel process computing like apache-spark,for huge amount of data we cant you tools like matlab. and we can compute most costly operating like finding the inverse and multiplying the large matrices efficiently by using better resources.

The results of the algorithm are plotted in results section.we can see that the error in the prediction is fallowing *skew-distribution.* and we can observe that the accuracy of prediction we less while increasing the depth of ocean i.e. we are getting more error. and finally from those results we can say that the accuracy in prediction of temperature is increased by these kind of algorithms.

Appendix A

KD-Tree



Nearest neighbor search is an important task which arises in different areas - pattern recognition, recommendation systems, DNA sequencing and even game development.

Usually, this task is formulated as follows. We have N points in some space (S dataset). We have to work with queries, which have dataset S and some point X as their parameters (X does not have to belong to S). Typical queries are "find k nearest neighbors of X" or "find all points in S at given distance R from X or closer". Depending on problem, we may have: a) different number of dimensions - from one to thousands, b) different metric type (Euclidean, 1-norm, ...), c) different dataset size. Hence, for different problems different algorithms are feasible.

The key point of the problem formulation is that dataset S is considered fixed. X may vary from request to request, but S remains unchanged. It allows to preprocess dataset and build data structure which accelerates processing. All strategies which promise better than O(N) processing time rely on some kind of preprocessing. Different preprocessing strategies have different features.

A.0.1 Nearest neighbor search with kd-trees

Kd-trees are data structures which are used to store points in k-dimensional space. As it follows from its name, kd-tree is a tree. Tree leafs store points of the dataset (one or several points in each leaf). Each point is stored in one and only one leaf, each leaf stores at least one point. Tree nodes correspond to splits of the space (axis-oriented splits are used in most implementations). Each split divides space and dataset into two

distinct parts. Subsequent splits from the root node to one of the leafs remove parts of the dataset (and space) until only small part of the dataset (and space) is left.

Chart at the left shows an example of kd-tree in the 2-dimensional space. Red squares are dataset points, black lines are splits. The thinner the line is, the deeper is the node which corresponds to the split.

kd-trees allow to efficiently perform searches like "all points at distance lower than R from X" or "k nearest neighbors of X". When processing such query, we find a leaf which corresponds to X. Then we process points which are stored in that leaf, and then we start to scan nearby leafs. At some point we may notice that distance from X to the leaf is higher than the worst point found so far. It is time to stop search, because next leafs won't improve search results. Such algorithm is good for searches in low-dimensional spaces. However, its efficiency decreases as dimensionality grows, and in high-dimensional spaces kd-trees give no performance over naive O(N) linear search (although continue to give correct results).

Considering number of dimensions K fixed, and dataset size N variable, we can estimate complexity of the most important operations with kd-tree:

	Avg Case	Worst Case
Space	$O(n)$	$O(n)$
Search	$O(\log n)$	$O(n)$
Insert	$O(\log n)$	$O(n)$
Delete	$O(\log n)$	$O(n)$

- **building a kd-tree** has $O(N \log N)$ time complexity and $O(KN)$ space complexity
- **nearest neighbor search** close to $O(\log N)$
- **M nearest neighbors** - close to $O(M \log N)$

Appendix B

Singular Values Decomposition

Given a complex matrix A having m rows and n columns, if σ is a non-negative scalar, and u and v are nonzero m- and n-vectors, respectively, such that

$$Av = \sigma u \; and \; A^{\intercal}u = \sigma v$$

then σ is a singular value of A and u and v are corresponding left and right singular vectors, respectively.(For generality it is assumed that the matrices here are complex, although given these results, the analogs for real matrices are obvious.)

If, for a given positive singular value, there are exactly t linearly independent corresponding right singular vectors and t linearly independent corresponding left singular vectors, the singular value has multiplicity t and the space spanned by the right (left) singular vectors is the corresponding right (left) singular space.

Given a complex matrix A having m rows and n columns, the matrix product $UV^{\intercal}$ is a singular value decomposition for a given matrix A if

- U and V, respectively, have orthonormal columns.
- Σ has non-negative elements on its principal diagonal and zeros elsewhere.
- $A = U\Sigma V^{\intercal}$.

Let p and q be the number of rows and columns of . U is $m \times p, p \leq m$, and V is $n \times q \; with \; q \leq n$.

There are three standard forms of the SVD. All have the i^{th} diagonal value of denoted σ_i and ordered as follows: $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_k$, and r is the index such that $\sigma_r > 0$ and either $k = r$ or $\sigma_r + 1 = 0$.

1. $p = mandq = n$. The matrix is $m \times n$ and has the same dimensions as A.

2. $p = q = minm, n$. The matrix Σ is square.

3. If $p = q = r$, the matrix is square. This form is called a **reduced SVD** and denoted by $\hat{U}\hat{\Sigma}\hat{v^{\intercal}}$

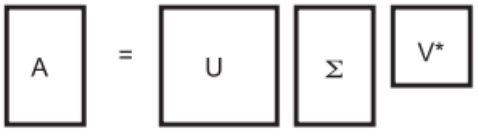


FIGURE B.1: The first form of the singular value decomposition where $m < n$.

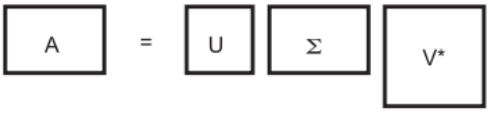


FIGURE B.2: The second form of the singular value decomposition where $m \geq n$.

FIGURE B.3: The second form of the singular value decomposition where $m < n$.

FIGURE B.4: The first form of the singular value decomposition where $m \geq n$.

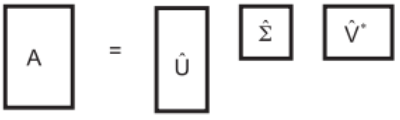


FIGURE B.5: The third form of the singular value decomposition where $r \leq n \leq m$

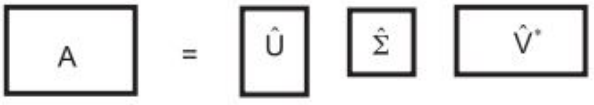


FIGURE B.6: The third form of the singular value decomposition where $r \leq n \leq m$

Appendix C

Data Source

We need three data file as input to optimal interpolation algorithm, the data consists of 2 variables a)Temperature b)Depth

C.0.2 Observational Data

Observational data taken from world wide database[5] it is normally of form [number of locations * number of depths] for every location data present at several depths.

The results in Section 6.2 are plotted on basis of daily data. the observational data is of size $[999 \times 15]$.

The results that are mentioned in section[section name] is monthly data comparatively has more locations and more depths for locations $[1577 \times 1146]$ than daily data.that after removing the NaN values we have observational vector of size [184833]. the observational points are considered for the Indian ocean region. from [70E-100E] longitude and [0N-24N] latitude region

C.0.3 True Data

true data, these are reference data values, the prediction of error is based on the true values. in this thesis true data is taken from the Asian Pacific atmospheric data center. the values are normally the best estimated values based on past several years of temperature at regular interval of girded points.

For daily data:

Resolution:0.5
Region: Indian ocean
Source: Objective Analysis
Longitude:76E to 100E
Latitude: 4N to 24N
Depths: [0-40]
Size: [longitudes * latitudes * depths] : [20*24*14]

For Monthly Data

Resolution:0.125
Region: Indian ocean
Source: Objective Analysis
Longitude:76E to 100E
Latitude: 4N to 24N
Depths: [0-1400]
Size: [longitudes * latitudes * depths] : [90*98*56]

C.0.4 Back Ground Data

back ground data, these are initial data values, based on the observations we add error to these values and predict assimilated values. in this thesis back ground values are taken from the output of objective analysis. the objective analysis is based on governing equations and estimates the temperature values at regular interval of girded points.

For daily data:

Resolution:0.5
Region: Indian ocean
Source: Asian Pacific atmospheric data center
Longitude:76E to 100E
Latitude: 4N to 24N
Depths: [0-40]
Size: [longitudes * latitudes * depths] : [20*24*14]

For Monthly Data

Resolution:0.125
Region: Indian ocean
Source: Asian Pacific atmospheric data center
Longitude:76E to 100E
Latitude: 4N to 24N
Depths: [0-1400]
Size: [longitudes * latitudes * depths] : [96*80*57]

Bibliography

[1] Atmospheric modeling, data assimilation and predictability by *Eugenia Kalnay* University of Maryland

[2] Data Assimilation Techniques,
`http://www.atmos.millersville.edu/~lead/Obs_Data_Assimilation.html`

[3] An easy-to-implement and efficient data assimilation method for the identification of the initial condition: the Back and Forth Nudging (BFN) algorithm *Didier Auroux1, Patrick Bansart2, Jacques Blum2*

[4] Asia Pacific Data Research Center, APDRC
`http://apdrc.soest.hawaii.edu/data/data.php?discipline_index=2`

[5] World Ocean Database Select and Search
`http://www.nodc.noaa.gov/OC5/SELECT/dbsearch/dbsearch.html`

[6] Weather Observations used in NWP.
`http://weather.mailasail.com/FranksWeather/Weather-Observations-Nwp`

[7] Numerical Weather Prediction and Data Assimilation *David Schultz, Mohan Ramamurthy, Erik Gregow, John Horel*

[8] Computation of the Singular Value Decomposition *Alan Kaylor Cline, Inderjit S. Dhillon*