Tsunami Prediction and Earthquake Parameters Estimation in the Red Sea

Thesis by

Zaid Sawlan

Submitted in Partial Fulfillment of the Requirements for the degree of Masters of Science

King Abdullah University of Science and Technology
Computer, Electrical and Mathematical Sciences and Engineering
Applied Mathematics and Computational Sciences

Thuwal, Makkah Province, Kingdom of Saudi Arabia

December, 2012
The thesis of Zaid Sawlan is approved by the examination committee.

Committee Chairperson: Ibrahim Hoteit
Committee Member: Marco Scavino
Committee Member: Taousmeriem Laleg
ABSTRACT

Tsunami concerns have increased in the world after the 2004 Indian Ocean tsunami and the 2011 Tōhoku tsunami. Consequently, tsunami models have been developed rapidly in the last few years. One of the advanced tsunami models is the GeoClaw tsunami model introduced by LeVeque (2011). This model is adaptive and consistent. Because of different sources of uncertainties in the model, observations are needed to improve model prediction through a data assimilation framework. Model inputs are earthquake parameters and topography. This thesis introduces a real-time tsunami forecasting method that combines tsunami model with observations using a hybrid ensemble Kalman filter and ensemble Kalman smoother. The filter is used for state prediction while the smoother operates smoothing to estimate the earthquake parameters. This method reduces the error produced by uncertain inputs. In addition, state-parameter EnKF is implemented to estimate earthquake parameters. Although number of observations is small, estimated parameters generates a better tsunami prediction than the model. Methods and results of prediction experiments in the Red Sea are presented and the prospect of developing an operational tsunami prediction system in the Red Sea is discussed.
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Illustrations</td>
<td>7</td>
</tr>
<tr>
<td>List of Tables</td>
<td>9</td>
</tr>
<tr>
<td><strong>I  Introduction</strong></td>
<td>10</td>
</tr>
<tr>
<td>I.1  Overview</td>
<td>10</td>
</tr>
<tr>
<td>I.2  Tsunami Modeling</td>
<td>11</td>
</tr>
<tr>
<td>I.3  The Red Sea</td>
<td>12</td>
</tr>
<tr>
<td>I.4  Objectives</td>
<td>14</td>
</tr>
<tr>
<td><strong>II  Tsunami Model</strong></td>
<td>15</td>
</tr>
<tr>
<td>II.1  Okada Model</td>
<td>15</td>
</tr>
<tr>
<td>II.2  Shallow Water Equations</td>
<td>16</td>
</tr>
<tr>
<td>II.2.1 1D shallow water equations</td>
<td>17</td>
</tr>
<tr>
<td>II.2.2 2D shallow water equations</td>
<td>18</td>
</tr>
<tr>
<td>II.3  Red Sea Tsunami Simulation</td>
<td>19</td>
</tr>
<tr>
<td><strong>III  Data Assimilation</strong></td>
<td>21</td>
</tr>
<tr>
<td>III.1  Kalman Filter</td>
<td>22</td>
</tr>
<tr>
<td>III.2  Extended Kalman Filter</td>
<td>23</td>
</tr>
<tr>
<td>III.3  Ensemble Kalman Filter (EnKF)</td>
<td>24</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>III.4</td>
<td>Ensemble Kalman Smoother (EnKS)</td>
</tr>
<tr>
<td>III.5</td>
<td>State-parameter Estimation with EnKF</td>
</tr>
<tr>
<td>IV</td>
<td>Implementation and Numerical Results</td>
</tr>
<tr>
<td>IV.1</td>
<td>Tsunami Prediction</td>
</tr>
<tr>
<td>IV.2</td>
<td>EnKF Results</td>
</tr>
<tr>
<td>IV.3</td>
<td>State-Parameters Estimation</td>
</tr>
<tr>
<td>V</td>
<td>Conclusions</td>
</tr>
<tr>
<td>Appendixes</td>
<td>45</td>
</tr>
<tr>
<td>A</td>
<td>EnKF Code</td>
</tr>
<tr>
<td>References</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Illustrations

I.1 Plate Tectonics of the Red Sea .................................................. 13

II.1 Red Sea Tsunami Simulation at different times .......................... 20

IV.1 Location of 30 buoys (black dots) in the red sea used in assimilation 32

IV.2 Comparison between true tsunami (left), model prediction (center) and tsunami after assimilation using $N_o = 30$ and $N_e = 30$ (right) at different times .................................................. 33

IV.3 Comparison between true tsunami (left), assimilation using EnKF with $N_o = 30$ and $N_e = 50$ (center) and assimilation using EnKF with $N_o = 50$ and $N_e = 50$ (right) at different times .................................................. 36

IV.4 Sea surface height from a single point in the red sea at location, 21.24° lat. and 38.14° lon. .................................................. 37

IV.5 Sea surface height from a single point in the red sea at location, 18.80° lat. and 39.80° lon. .................................................. 37

IV.6 Earthquake parameters estimation where $T = 120$ minutes .......... 39

IV.7 Earthquake parameters estimation where $T = 240$ minutes .......... 40

IV.8 Comparison between true tsunami (left), model prediction using smoothed parameters $\tilde{p}$ (center) and assimilation using EnKF and smoothed parameters (right) at different times .................................................. 41
<table>
<thead>
<tr>
<th>IV.9</th>
<th>Comparison between true tsunami, model prediction using smoothed parameters ( \tilde{p} ) and assimilation using EnKF and smoothed parameters at the point 15.20° lat. and 41.80° lon.</th>
<th>42</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV.10</td>
<td>Comparison between true tsunami, model prediction using smoothed parameters ( \tilde{p} ) and assimilation using EnKF and smoothed parameters at the point 18.80° lat. and 39.80° lon.</td>
<td>42</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1</td>
<td>earthquakes history [2]</td>
<td>13</td>
</tr>
<tr>
<td>IV.1</td>
<td>True parameters of the 1977 earthquake in the red sea.</td>
<td>31</td>
</tr>
</tbody>
</table>
Chapter I

Introduction

I.1 Overview

Natural hazards have serious impacts on people, environment and economy. One of the natural hazards that can cause a loss of thousands lives is tsunami. A tsunami is a series of waves created by an underwater disturbance such as an earthquake, landslide, or volcanic eruption [3]. For example, the 2004 Indian Ocean tsunami affected several countries and caused the death of more than 200,000 people [4]. Just recently, the 2011 Tōhoku tsunami affected some nuclear plants in Japan and thousands of people died. Both tsunamis were caused by earthquakes with an approximate magnitude 9 $M_w$ [5]. This work will only consider tsunamis generated by earthquakes which are the most common cause for tsunamis [3].

Tsunami warnings are very important to save life of people in coastal regions. Based on these warning, local authorities can immediately evacuate people and decrease the chances of inundation. Actions can be helpful and useful if warnings are given as early and as accurate as possible. Tsunami warning systems use seismic
waves created by earthquakes with a tsunami model as indication of tsunami generation. However, earthquake parameters obtained by seismic waves are often poorly known leading to important uncertainties in the model prediction of the tsunami. In general, tsunami models are subject to two sources of errors and uncertainties, which strongly limit the accuracy of tsunamis prediction. First source of errors comes from model inputs namely the bathymetry (sea topography) and earthquake parameters. The second source of errors is due to numerical approximation used for implementation in the model. Earthquake parameters are the most important source of errors since there is no prior knowledge of them before tsunami. These errors and uncertainties can be reduced by tsunami observations. Buoys devices in the ocean provide tsunami measurements with some noises. In some cases, satellite also can provide observations closer to the earthquake source. However, some satellite observations are not related to tsunami and could be caused by wind and eddies [6]. On the other hand, coastal tide gauges give accurate information about tsunami waves but do not give enough time for evacuation of the population. Therefore, we need a tsunami forecasting method that combines tsunami measurements with a tsunami model to have a good and early prediction of tsunamis [7, 8]. This process of combining models and data is referenced in the atmospheric and oceanic communities as data assimilation [9]. This work will use a data assimilation methodology based on ensemble Kalman filter to study tsunami prediction in the Red Sea. The presented approach is however general and easily implemented to any other region with risk of tsunamis.

I.2 Tsunami Modeling

After the 2004 Indian Ocean tsunami, tsunami modeling expanded quickly and many researches developed new codes. For example, Zhang and Batista (2009) presented an
example of the adaptation of a multi-purpose circulation model (SELFE) to tsunami inundation and propagation [10]. Another example was introduced by LeVeque (2011). It is implemented in GeoClaw, open source software that is available as part of Clawpack [11]. This model uses adaptive finite volume methods and it was used to show tsunami generated by the 27 February 2010 earthquake near Maule, Chile. The last model will be considered in this research and will be presented with more details in chapter [11].

Pacific Marine Environmental Laboratory (PMEL) has developed different types of models that combines real-time data with numerical model. This model requires that the solution must provide the best fit to the observations within specific times. The inversion problem of fitting data is solved by using a discrete set of Greens functions [7]. PMEL model has been used by tsunami warning centers in Hawaii and Alaska [12, 13]. Another approach to combine observations with tsunami model is using ensemble Kalman filter and smoother. This will be the focus of our study and will be discussed extensively in chapter (III).

I.3 The Red Sea

In this research the tsunami model will be implemented in the Red Sea. Therefore, a general overview of the structure of the Red Sea is given. The Red Sea lies in a fault depression between the Arabian plate and the Nubian plate [14] and both plates meet with the Somalian plate in the south [1] as in Figure (I.1). As a result of the motion that happens between these three plate, earthquakes sometimes occur in the Red Sea. A table of some earthquakes history in the Red Sea is provided with some parameters (I.1). For example, the fault plane of 1977 earthquake was oriented at
106 degrees from north (Strike direction) and dipped at 66 degrees from horizontal (Dip angle). The slip was in the direction -171 degrees from the strike direction (Slip angle). These earthquakes can generate tsunamis that affect coastal cities and ships cruising in the Red Sea.

Table I.1: earthquakes history [2]

<table>
<thead>
<tr>
<th>Year</th>
<th>Magnitude</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Depth</th>
<th>Strike</th>
<th>Dip</th>
<th>Slip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1977</td>
<td>6.6</td>
<td>15.97</td>
<td>40.32</td>
<td>10.1</td>
<td>106</td>
<td>66</td>
<td>-171</td>
</tr>
<tr>
<td>1980</td>
<td>6.0</td>
<td>17.12</td>
<td>40.53</td>
<td>15.0</td>
<td>24</td>
<td>76</td>
<td>-9</td>
</tr>
<tr>
<td>1988</td>
<td>5.6</td>
<td>16.56</td>
<td>41.10</td>
<td>15.0</td>
<td>339</td>
<td>74</td>
<td>-17</td>
</tr>
<tr>
<td>1993</td>
<td>5.6</td>
<td>19.42</td>
<td>38.55</td>
<td>15.0</td>
<td>144</td>
<td>40</td>
<td>-84</td>
</tr>
<tr>
<td>1996</td>
<td>5.3</td>
<td>19.13</td>
<td>38.94</td>
<td>15.0</td>
<td>153</td>
<td>20</td>
<td>-68</td>
</tr>
<tr>
<td>2001</td>
<td>5.2</td>
<td>18.21</td>
<td>40.07</td>
<td>15.0</td>
<td>303</td>
<td>36</td>
<td>-128</td>
</tr>
<tr>
<td>2006</td>
<td>4.7</td>
<td>19.09</td>
<td>39.28</td>
<td>12.0</td>
<td>329</td>
<td>29</td>
<td>-93</td>
</tr>
<tr>
<td>2010</td>
<td>5.0</td>
<td>18.72</td>
<td>39.46</td>
<td>17.5</td>
<td>329</td>
<td>29</td>
<td>-59</td>
</tr>
</tbody>
</table>
I.4 Objectives

The main objective of this thesis is to improve models prediction of tsunamis ahead of time when earthquakes happen using data assimilation. The prediction will be produced using a deterministic tsunami model but the earthquake parameters will be treated as unknowns and random. Ensemble Kalman filter will be used to assimilate observations to the model. This work is the first attempt to use data assimilation methods for tsunami prediction. Another objective is to recover the earthquake parameters which are the inputs of the tsunami model. These parameters have uncertainties which can be reduced by the observations obtained from buoys in the sea. This problem can be viewed as an inverse problem which will be solved by a new method that combines filtering with smoothing. This method will be called hybrid ensemble Kalman filter and ensemble Kalman smoother.

The thesis is organized as follows:

Chapter II presents the tsunami model we will use and its implementation and configuration in the Res Sea. Chapter III describes the ensemble Kalman filter and smoother and how to use observations from the sea with the model in order to improve prediction accuracy. This chapter will also introduce an approach for state-parameter estimation based on a hybrid ensemble Kalman filter for state estimation and ensemble kalman smoother for parameter estimation. Chapter IV will present an experiment setup and numerical results. The results will show how tsunami prediction is improved by EnKF based on observations and how estimated parameters can make a better model prediction.
Chapter II

Tsunami Model

This chapter elaborates the tsunami model used in this research. The model can be divided into two main parts. The first part is Okada model which transform earthquakes to surface deformation. The second part is simulating the tsunami by solving the shallow water equations. This tsunami model is available online with full documentation as part of the Clawpack software [15].

II.1 Okada Model

The Okada model calculates analytical solution for surface deformation due to shear and tensile faults in an elastic half-space. This model is commonly used to simulate ground deformation produced by earthquakes [16]. The model only requires the earthquake parameters and the topography of modeled area to computes surface deformation. Topography data of any region can be found at the National Geophysical Data Center (NGDC) [17]. The earthquake parameters are the rectangular fault geometry (length, width, depth, strike, dip), the magnitude (dislocation, slip) and the epicenter of the earthquake (longitude, latitude) [18].
II.2 Shallow Water Equations

The two-dimensional shallow water equations are usually used to model tsunamis. In this model, it is possible to use adaptive mesh refinement to compute accurate solutions. However, only the uniform mesh is used in this preliminary work to avoid coding complications that might be caused by adaptive meshes. Future work will consider an adaptive mesh.

The shallow water equations are a nonlinear hyperbolic system of partial differential equations (conservation laws for depth and momentum) and solutions may contain shock waves. Therefore, shock-capturing methods that can converge to discontinuous weak solutions should be used. The two-dimensional shallow water equations:

\[
\begin{align*}
    h_t + (hu)_x + (hv)_y &= 0, \\
    (hu)_t + (hu^2 + \frac{1}{2}gh^2)_x + (huv)_y &= -ghB_x, \\
    (hv)_t + (huv)_x + (hv^2 + \frac{1}{2}gh^2)_y &= -ghB_y,
\end{align*}
\]  

(II.1)

where \( h(x, y, t) \) is the fluid depth, \( u(x, y, t) \) and \( v(x, y, t) \) are depth-averaged velocities in the two horizontal directions, \( g \) is the gravitational constant and \( B(x, y, t) \) is the topography. In practice, topography is assumed to be constant \( B(x, y, t) = B(x, y) \), therefore, fluid depth can be replaced by fluid surface height.
II.2.1 1D shallow water equations

For simplicity, the one dimension form of the shallow water equations will be discussed first.

\[ h_t + (hu)_x = 0, \quad (II.2) \]
\[ (hu)_t + (hu^2 + \frac{1}{2}gh^2)_x = -ghB_x, \quad (II.3) \]

which can be written in the general form of hyperbolic systems as following

\[ q_t + f(q)_x = \psi(q, x), \quad (II.4) \]

where \( q(x, t) \) is the vector of unknowns, \( f(q) \) is the vector of corresponding fluxes and \( \psi(q, x) \) is a vector of source terms:

\[ q = \begin{bmatrix} h \\ hu \end{bmatrix}, \quad f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad \psi = \begin{bmatrix} 0 \\ -ghB_x \end{bmatrix}, \quad (II.5) \]

and the Jacobian matrix:

\[ f'(q) = \begin{pmatrix} 0 & 1 \\ gh - u^2 & 2u \end{pmatrix}. \quad (II.6) \]

A class of numerical methods that can be used to solve the system (II.4) is Godunov-type methods [19]. The algorithm of this method can be briefed as following:

\[ Q_i^n = \frac{1}{V_i} \int_{c_i} q(x, t_n) dx, \quad (II.7) \]
\[ Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left( A^+ \Delta Q_{i-1/2}^n + A^- \Delta Q_{i+1/2}^n \right), \quad (II.8) \]
where $Q_i^n$ is an approximation to the average value of the solution in the ith grid cell $C_i$, $V_i$ is the volume of the grid cell, $A^+ \triangle Q_{i-1/2}^n$ and $A^- \triangle Q_{i+1/2}^n$ are the net effects of all waves propagating into the cell from the left and right boundary, respectively.

Using equation (II.8) will give first order accurate solutions. To increase the resolution, correction terms must be added to (II.8). For example, the Lex-Wendroff method is second order accurate and it is given by:

$$Q_{i}^{n+1} = Q_i^n - \frac{1}{2} \frac{\Delta t}{\Delta x} A(Q_{i+1}^n - Q_{i-1}^n), + \frac{1}{2} \left( \frac{\Delta t}{\Delta x} \right)^2 A^2(Q_{i+1}^n - Q_i^n + Q_{i-1}^n).$$ (II.9)

Finally, applying limiters to (II.9) will produce high resolution results [19].

### II.2.2 2D shallow water equations

In two space dimensions, the general form of hyperbolic systems (II.4) is written as following:

$$q_t + f(q)_x + g(q)_y = \psi(q, x, y).$$ (II.10)

Godunov-type finite volume algorithms can be extended to two dimensions simply by solving 1D problems normal to each edge of a finite volume cell $C_{ij}$. In this case the numerical solution $Q_{ij}^n$ is an approximation to the average value of the solution over the grid cell $C_{ij}$

$$Q_{ij}^n = \frac{1}{V_{ij}} \int_{C_{ij}} q(x, y, t_n) dx dy,$$ (II.11)

where $V_{ij}$ is the area of the cell.

High-resolution correction terms can then be added to achieve greater accuracy without spurious oscillations. The methods used in GeoClaw are the standard wave
II.3 Red Sea Tsunami Simulation

An example of tsunami simulation in the Red Sea can be modeled using 1977 earthquake parameters in Table (I.1). The bathymetry was obtained from the ETOPO2 data set at the National Geophysical Data Center (NGDC) and it is 2 minutes resolution in latitude and longitude. Figure (II.1) shows some plots of tsunami at different times. First plot shows two waves were generated after 5 minutes of the earthquake. One wave is about 2 meters height and it is moving to north as can be seen in the second plot after half an hour. After one hour, the waves almost reached coasts but with surface height around one meter. In the plot, we see tsunami waves were reflected from the coasts and started to dissipate.
Figure II.1: Red Sea Tsunami Simulation at different times

(a) 5 minutes

(b) 30 minutes

(c) One hour

(d) One hour and 30 minutes
Chapter III

Data Assimilation

Data assimilation is the process of combining information from uncertain predictive models and noisy observations to determine the best possible description of a dynamic system [9]. The presence of observational data is essential as it helps in guiding the model towards true and realistic trajectories, especially in cases where our knowledge of the physics in the system is limited and fairly inaccurate. Assimilation methods can be classified into two main categories; sequential schemes known for their recursive online updating feature and variational schemes where optimization is applied at once for the entire system using all available observations [9]. The work carried out in this thesis falls within the sequential category and is based on the well-known Kalman Filter (KF). As it will be illustrated in the following sections, the KF can be only used for linear models with certain physical and probabilistic assumptions such as the Gaussianity of the distributions. In case of nonlinear models, however, other variants of the KF are used and from these we mention the Extended Kalman Filter (EKF), the Ensemble Kalman Filter (EnKF) and the Ensemble Kalman Smoother (EnKS). These filters differ in terms of their mathematical derivation and statistical formulation. In all assimilation methods, the following discrete dynamic and time
processes are considered:

\begin{align*}
x_k &= \mathcal{M}_k(x_{k-1}) + v_k, \quad (\text{III.1}) \\
y_k &= H x_k + w_k, \quad (\text{III.2})
\end{align*}

where $x_k$ is the dynamic state of the system at time $t_k$, $y_k$ is the observational vector available at time $t_k$, $\mathcal{M}_k$ is dynamic operator integrating the system state from $t_{k-1} \rightarrow t_k$ and $H$ is a linear observational operator mapping the state variables from $\mathbb{R}^N \rightarrow \mathbb{R}^{N_o}$. Here, $N$ is the dimension of the state vector and $N_o$ is the dimension of the observation space. $v_k$ and $w_k$ are the respective model and observation noise vectors; assumed to be Gaussian with zero means and covariance matrices $Q_k$ and $R_k$, respectively.

### III.1 Kalman Filter

The KF is a statistical data assimilation scheme that provides the best estimate, in the sense of minimum variance, of the state of a linear system with Gaussian errors using all observations up to the estimation time \cite{20}. The body of the KF can be described by a set of recursive mathematical equations where its optimality criteria relies on minimizing the mean squared estimation error. The KF is very powerful in handling poorly known model parameters and initial conditions by providing not only state estimates but also the underlying uncertainties. Sequentially, the KF integrates the state and its statistics in time and then it updates the model prediction every time new observations are available. The algorithm can be briefly summarized as

- Forecast (propagation) step: At time $t_{k-1}$, the state of the system has a mean $x_{k-1}^a$ and covariance $P_{k-1}^a$ and then propagation to the next time step takes
place as follows:

\[
\begin{align*}
x_k^f &= \mathcal{M}_k x_{k-1}^a, \\
P_k^f &= \mathcal{M}_k P_{k-1}^a \mathcal{M}_k^T + Q_k,
\end{align*}
\] (III.3)

here \( \mathcal{M}_k \) is a linear modeling operator.

- Analysis (update) step: Once observations are available at time \( t_k \), model prediction is updated in a linear fashion as

\[
\begin{align*}
x_k^a &= x_k^f + K_k (y_k - H_k x_k^f), \\
P_k^a &= (I - K_k H_k) P_k^f, \\
K_k &= P_k^f H_k^T (H_k P_k^f H_k^T + R_k)^{-1},
\end{align*}
\] (III.5)

where \( K_k \) denotes the Kalman gain matrix acting as a weighting term between the model prediction and observations.

Prior to these 2 steps, the Kalman filter is initialized with a mean state \( x_0^a \) and covariance \( P_0^a \).

### III.2 Extended Kalman Filter

In case of nonlinear dynamics, the KF can no longer be used as the assimilation tool because it is only designed for linear models. To overcome this, Aurther Gelb in 1974 proposed a modified version of the KF to deal with nonlinear models. The method involves linearization of the operator \( \mathcal{M} \) around the current state of the system at every prediction cycle. The linearized model is then used to propagate the error covariance matrix to the next time step. The linearization is done by expanding the
term $\mathcal{M}_k (x_{k-1})$ using Taylor series expansion:

$$\mathcal{M}_k (x_{k-1}) = M_k (x_{k-1}) + \nabla M_k (x_{k-1} - x_{k-1}^a) + \mathcal{O} \left(|x_{k-1} - x_{k-1}^a|^2\right), \quad (\text{III.8})$$

$$x_k \approx M_k (x_{k-1}) + \nabla M_k (x_{k-1} - x_{k-1}^a) + v_k, \quad (\text{III.9})$$

where $\nabla M_k = (\partial M_k / \partial x_{k-1}) |_{x_{k-1}^a}$ denotes the gradient (Jacobian) of the model. This implies that $x_k \sim \mathcal{N} \left(x_k^f, P_{k}^f\right)$, with

$$x_k^f = M_k (x_{k-1}^a), \quad (\text{III.10})$$

$$P_{k}^f = \nabla M_k P_{k-1}^a \nabla M_k^T + Q_k. \quad (\text{III.11})$$

Once the forecast is done, the analysis step is then similar to the linear KF one. The EKF - unlike the KF- has no optimality properties and no direct probabilistic interpretation. Moreover, the EKF is computationally expensive to apply because for every forecast, it requires $N + 1$ model runs to approximate the gradient of the model $\nabla \mathcal{M}$ with simple forward differences. This sometimes might cause dynamic instabilities if the dimension of the system is very large [21, 22]. As a remediation for these computational and numerical drawbacks, the EnKF was introduced by Evensen in 1994.

### III.3 Ensemble Kalman Filter (EnKF)

The EnKF is a Monte Carlo based algorithm that deals with samples rather than distributions or full covariance matrices. The EnKF assumes a random sample or ensemble

$$\{x^i, i = 1, 2, ..., N_e\}, \quad (\text{III.12})$$
rather than one single state vector where \( N_e \) is the number of ensemble members.

At every forecast stage, all ensemble members are integrated with the nonlinear dynamic model and then a sample covariance matrix is computed without any need for linearization as follows

\[
x^{f,i}_k = M_k (x^{a,i}_{k-1}) ,
\]

\[
\hat{P}^f_k = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (x^{f,i}_k - \bar{x}^f_k) (x^{f,i}_k - \bar{x}^f_k)^T ,
\]

where \( \bar{x}^f_k \) is the sample mean of the propagated ensemble members computed as shown below

\[
\bar{x}^f_k = \frac{1}{N_e} \sum_{i=1}^{N_e} x^{f,i}_k .
\]

The sample forecast covariance need not to be fully formed (if the dimension is too large); instead it is decomposed into ensemble anomalies as

\[
\hat{P}^f_k = E_{x_k} E_{x_k}^T ,
\]

where the columns of \( E_{x_k} \) are \( (N_e - 1)^{-\frac{1}{2}} (x^{f,i}_k - \bar{x}^f_k) \).

In the analysis step, the ensemble members are updated using the linear KF update equation

\[
x^{a,i}_k = x^{f,i}_k + \hat{K}_k [ y_k - H_k f^{f,i}_k ] ,
\]

where \( \hat{K}_k \) is the approximate Kalman gain at \( t_k \) computed as

\[
\hat{K}_k = \hat{P}_{xy_k} (\hat{P}_{yy_k} + R_k)^{-1} ,
\]
where the covariance terms $\hat{P}_{xy}^f$ and $\hat{P}_{yy}^f$ are defined as

$$
\hat{P}_{xy}^f = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (x_{k}^{f,i} - \bar{x}^f_k) \left( Hx_{k}^{f,i} - H\bar{x}^f_k \right)^T \equiv E_{x_k} E_{y_k}^T, \quad (III.19)
$$

$$
\hat{P}_{yy}^f = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} \left( Hx_{k}^{f,i} - H\bar{x}^f_k \right) \left( Hx_{k}^{f,i} - H\bar{x}^f_k \right)^T \equiv E_{y_k} E_{y_k}^T, \quad (III.20)
$$

One important feature of the EnKF is the treatment of the observations as random variables with mean equal to the actual observation and a predefined covariance where

$$
y_k^i = y_k + \epsilon_k^i, \quad (III.21)
$$

and $\epsilon_k^i \sim \mathcal{N}(0, R_k)$.

The accuracy of EnKF depends on the ensemble size $N_e$ and by increasing $N_e$, the estimation is expected to improve as $\bar{x}^f_k$ and $\hat{P}_{xy}^f$ will converge to $E(x_k^f)$ and cov($x_k^f$) but this will; however, increase the complexity of the computation. Actually, if the cost of running the model is $O(M)$, then the cost of EnKF is $O(N_e M)$. EnKF was implemented in Matlab and illustration of the code is available in appendix A.

### III.4 Ensemble Kalman Smoother (EnKS)

Ensemble Kalman Smoother (EnKS) consists of a forward pass and a backward pass. First, the EnKF in run forward in time for $k = 0, ..., T$ storing the forecast and filtered ensembles at each time step. Then a backward pass is implemented to obtain the smoothed ensemble. In other words, the ensemble states at $k = t$ are updated using all future observations where $k > t$ [23]. The smoother is initialized at the final time $T$ by setting $x_T^s,i = x_T^{a,i}$. Then, we proceed backward in time using the following
recursive updating equation:

\begin{align*}
x^{s,i}_k &= x^{a,i}_k + B_k \left( x^{s,i}_{k+1} - x^{f,i}_{k+1} \right), \quad \text{(III.22)} \\
B_k &= P_k^a M'_k \left( P_k^f \right)^{-1}. \quad \text{(III.23)}
\end{align*}

where $M'$ is the linearized evolution matrix \[24\].

This smoother is commonly used to estimate initial conditions of the state. In the tsunami model, earthquake parameters are just initial conditions of the model. However, earthquake parameters are not time-variant with model so this smoother can not be used to estimate them. The goal now is to modify this smoother such that it estimates the initial parameters. This modified smoother is introduced in the next section.

### III.5 State-parameter Estimation with EnKF

Assimilation methods discussed above focused on state estimation problems. This could be enough for applications with no real emergency. Tsunami applications, however, require estimating model parameters alongside with the state variables. Poor knowledge of the modeling parameters, especially in our earthquake-tsunami application, will lead in most of the cases to inaccurate state estimates. These physical parameters can be time variant or just constants. One approach for the combined estimation problem is given by the joint estimation where the states and the parameter vectors are added together in a single joint state vector; commonly referred to as state augmentation approach \[25\] \[26\]. The parameters are then updated with state observations using cross-correlations terms of the joint covariance matrix. At every
forecast step, the new updated parameters are fed to the model as follows

\[ x_{f,i}^k = \mathcal{M}_k \left( x_{a,i}^{k-1}, p_{a,i}^{k-1} \right), \]  
(III.24)

where \( p_{a,i}^{k-1} \) is the updated parameter ensemble of size \( N_p \times N_e \). Note that if the parameters are time invariant, then they are propagated assuming no dynamics according to

\[ p_{f,i}^k = p_{a,i}^{k-1}. \]  
(III.25)

The update equation for the state variables and the parameters is then

\[ Z_{a,i}^k = Z_{f,i}^k + \hat{K}_k \left( y^i - \mathcal{H}_k Z_{f,i}^k \right), \]  
(III.26)

where the joint vector is formed as \( Z_k = \begin{bmatrix} x_k & p_k \end{bmatrix}^T \). The new observational operator, \( \mathcal{H} \) has additional \( N_p \) zero columns to eliminate the parameters from the observation space.

For the considered tsunami model, earthquake parameters represent only initial input so the formulated EnKF cannot be used as it is and requires more adjustments. After the first assimilation, say at \( k = 1 \), we obtain an updated ensemble of parameters \( \tilde{p}_i^1 \) that should be used to initialize the model again. In order to use the second observation, we forecast the model using \( \tilde{p}_i^1 \) up to \( k = 2 \), then we assimilate to obtain a new \( \tilde{p}_i^2 \) and continue the same way. This modified joint state-parameter estimation with the EnKF method can be briefly summarized with the following algorithm:

1. Initialize model using some parameter vector.

2. Apply an EnKF cycle for the joint state at \( k = 1 \).

3. Set \( k = k + 1 \).
4. Re-initialize model using the updated parameters.

5. Apply state-parameters EnKF at $k$.

6. Go to step 3.

The algorithm stops when $k = T$ and it gives smoothed earthquake parameters $\tilde{p}$ using all observation $k = 1, 2, ..., T$. Final time $T$ must be chosen carefully in order to improve the estimation and be able to predict tsunami upon time. Computationally, the cost of this method is $O((NM)^2)$ since the model is running from the beginning for each iteration. This new method can be called hybrid EnKF-EnKS since filtering was used for state estimation while smoothing was used for parameter estimation.
Chapter IV

Implementation and Numerical Results

In this section, we apply the data assimilation schemes on state and state-parameter estimation problems for the tsunami prediction in the Red Sea. One of our goals is to assess whether a filtering approach is enough to provide accurate predictions starting from poor earthquake parameters. Our state variables represent the sea surface and the momentum in x and y directions ($N \approx 271000$ variables). Assimilation experiments are ran assuming realistic earthquake parameter values; i.e. $\hat{p} = p + \delta$ where $\delta$ is an additive noise accounting for seismic measurement error. These errors result, in most of the cases, because of inexact machine precision that is delivering the data or due to human reading mistakes. We believe that the choice of this noise $\delta$ highly affects the performance of the model and the assimilation schemes. Two scenarios for the assimilation experiments were considered as follows:

First, we assume that the parameters are perturbed and EnKF is applied to recover the tsunami in the sea.

Second, we work on minimizing norm of the noise; in other words, recover the true
parameters $p$ using the joint state-parameter approach with EnKF discussed before.

The true 9 parameters of $p$ (listed in table IV.1), are taken as those of the 1977 earthquake in the Red Sea. These parameters are then perturbed with some noise $\delta$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault width</td>
<td>130 km</td>
</tr>
<tr>
<td>Fault length</td>
<td>300 km</td>
</tr>
<tr>
<td>Slip</td>
<td>-101°</td>
</tr>
<tr>
<td>Dip angle</td>
<td>70°</td>
</tr>
<tr>
<td>Strike angle</td>
<td>96°</td>
</tr>
<tr>
<td>Dislocation</td>
<td>17 m</td>
</tr>
<tr>
<td>Latitude</td>
<td>15.97°</td>
</tr>
<tr>
<td>Longitude</td>
<td>40.32°</td>
</tr>
<tr>
<td>Depth</td>
<td>13 km</td>
</tr>
</tbody>
</table>

Table IV.1: True parameters of the 1977 earthquake in the red sea.

taken as Gaussian with the following values:

$$\delta = [-35, -50, -40, -60, 84, 2, -1.09, 0.77, 1]^T.$$ \hfill (IV.1)

We propose two observational networks where the number of buoys in the sea is: $N_o = 30$ and $N_o = 50$. In both cases, the buoys are distributed vertically in the Red Sea as seen in figure IV.1. Numerical results were compared for two ensemble sizes; $N_e = 30$ and $N_e = 50$. For $N_e > 50$, running EnKF becomes computationally very expensive and sometimes impossible to carry out using the computational resources that were made available for this project. The total simulation time is 90 minutes and observations are assimilated every 5 minutes.
IV.1 Tsunami Prediction

In the first scenario, we compare the performance of a model free-run (without assimilation) with perturbed parameters $\hat{p}$ to the reference "true" run. We observe that the model prediction gets degraded quickly after 5 minutes showing smaller and shorter spread of sea surface heights as compared to the true ones (figure IV.2). As time passes, model predictions continue to provide inaccurate tsunami surface heights as seen from the figure after 30, 60 and 90 minutes. This suggests that using uncertain model parameters highly affects the predictions of the model. With this configuration, model outputs are not reliable that is why observational data are very essential to assist the model in providing better estimates.

IV.2 EnKF Results

In this section, we test the same first scenario with the EnKF in order to improve the estimates. Results from three cases are discussed based on the number and
Figure IV.2: Comparison between true tsunami (left), model prediction (center) and tsunami after assimilation using $N_o = 30$ and $N_e = 30$ (right) at different times.
distribution of observations, \( N_o \) and the chosen ensemble size, \( N_e \).

- **Case 1**: \( N_o = 30 \) and \( N_e = 30 \)
  Results for this run are shown in figure [IV.2]. The spread of the tsunami heights are closer to the reference run than model predictions. This is seen for the whole 90 minutes window suggesting the important usage of the observations within an EnKF.

- **Case 2**: \( N_o = 30 \) and \( N_e = 50 \)
  As illustrated before (in the data assimilation chapter), the EnKF estimates are expected to improve when the ensemble size gets larger. In figure [IV.3], the tsunami surface heights are shown using case 2 configuration in the center. It’s not easy to decide whether increasing the ensemble size has improved the results so we propose to plot the height at a single point in the red sea and compare the performance of the filters. In figures [IV.4] and [IV.5], we see that the surface height is better estimated with \( N_e = 50 \); however, still far from the true reference solution.

- **Case 3**: \( N_o = 50 \) and \( N_e = 50 \)
  In this case, we increase the number of buoys in the red sea from 30 to 50 making the observational network in figure [IV.1] more dense (with similar distribution). The 2D plot of the whole surface height is shown again on the right side of figure [IV.3] but for viewing reasons we look at the surface height from the 2 points in figures [IV.4] and [IV.5]. Clearly, we observe that assimilating more observations has improved the results and made them closer to the reference one especially at the second point located at the center of the tsunami.

With these 3 cases, we’ve seen that assimilating data is very important to assist model predictions and get more accurate estimates. The EnKF performed well and
exhibited some sensitivity to the chosen number of buoys within the domain. Also, choosing larger ensemble size guarantees better results. We believe that increasing the assimilation window can improve the EnKF estimates but the red sea region is very small and so, the duration of the tsunami is very short.

IV.3 State-Parameters Estimation

In the second scenario, earthquake parameters are estimated simultaneously with the state variables by applying the modified joint state-parameter EnKF to tsunami model as explained in Chapter III. Figure (IV.6) shows the estimated parameters $\tilde{p}$ in time using observations up to $T = 120$ minutes as compared to the true parameters $p$. The most important parameters to look at are the slip, dislocation, latitude and longitude. These 4 parameters mostly determine the strength and the location of the earthquake. Clearly with 50 observations and 271000 state variables, the system is highly underdetermined and the parameters are not expected to be fully recovered. From the figure, we see that the estimated parameters are not accurately recovered but all of them were better at the end of the window than the initial ones. Another experiment was further implemented to check the effect of increasing the assimilation window on the goodness of the parameters. It is important to mention that the tsunami mostly happens in the first two hours after the earthquake; therefore, not much improvement is expected if assimilation continues for extra 2 hours. In figure [IV.7], we assimilated data up to 4 hours after the earthquake stroked the red sea. Nicely, the parameters beyond 2 hours were almost the same reflecting the fact that the tsunami is over and further improvement in their estimation will not happen.

Moreover, a new prediction for the tsunami can be made using the estimated (smoothed) parameters $\tilde{p}$ resulted from assimilation. In figure [IV.8], we can clearly
Figure IV.3: Comparison between true tsunami (left), assimilation using EnKF with $N_o = 30$ and $N_e = 50$ (center) and assimilation using EnKF with $N_o = 50$ and $N_e = 50$ (right) at different times
Figure IV.4: Sea surface height from a single point in the red sea at location, 21.24° lat. and 38.14° lon.

Figure IV.5: Sea surface height from a single point in the red sea at location, 18.80° lat. and 39.80° lon.
see the similarity between the tsunami simulated by $\tilde{p}$ and the reference one. The waves of the reference tsunami; however, are higher and this is also seen in figures (IV.9) and (IV.10). To improve the new tsunami prediction, we implement ensemble Kalman filter to estimate the sea surface. Figures (IV.8), (IV.9) and (IV.10) show a good approximation for the tsunami using EnKF with smoothed parameters.
Figure IV.6: Earthquake parameters estimation where $T = 120$ minutes.
Figure IV.7: Earthquake parameters estimation where $T = 240$ minutes.
Figure IV.8: Comparison between true tsunami (left), model prediction using smoothed parameters \( \tilde{p} \) (center) and assimilation using EnKF and smoothed parameters (right) at different times.
Figure IV.9: Comparison between true tsunami, model prediction using smoothed parameters $\hat{p}$ and assimilation using EnKF and smoothed parameters at the point 15.20° lat. and 41.80° lon.

Figure IV.10: Comparison between true tsunami, model prediction using smoothed parameters $\hat{p}$ and assimilation using EnKF and smoothed parameters at the point 18.80° lat. and 39.80° lon.
Chapter V

Conclusions

In summary, we first implemented EnKF to the tsunami model with different scenarios. Then, earthquake parameters were estimated using the hybrid EnKF-EnKS. Finally, EnKF was used again for tsunami prediction with the estimated parameters. Based on the results we conclude the following:

Tsunami prediction can be improved by ensemble Kalman filter and it provides real-time forecasting. If earthquake parameters are very noisy, then EnKF needs a large number of observations. Observations can be obtained from buoys, tide gauges and from satellite if possible. In case of lack of observations, other techniques can be used such as localization [27]. Also, EnKF results improve with time as number of assimilation increases. Since the Red Sea is narrow, the life time of tsunamis is very short compared to tsunamis in open oceans. As a result, providing timely accurate predictions in the Red sea becomes very challenging.

For parameters estimation, we need to increase number of ensembles to compute more reliable correlations between states and parameters. Moreover, increasing num-
ber of observations is helpful specially when state vector is too large. In order to use estimated parameters for tsunami prediction, final time $T$ must be smaller than arrival time with at least half an hour otherwise warnings will not be effective. Computationally, the hybrid EnKF-EnKS requires $O(N^2_e)$ model runs. To reduce time of running the model, different ensembles can be executed in parallel. We also found that state estimation approach is not enough because of the sparsity of the data and the short available time to provide timely forecasts. Much better results were obtained when earthquake parameters were estimated.
Appendix A

EnKF Code

function [avr, Xka] = enkfa(H,Xkf,Yk,R)

[n,q] = size(Xkf); %q = number_ensembles, %n = dimension of state
p = length(Yk); %p = dimension of observations

Ykf = H*Xkf;
xkf_bar = mean(Xkf,2);
Ekfx = Xkf - xkf_bar*ones(1,q);
ykf_bar = mean(Ykf,2);
Ekfy = Ykf - ykf_bar*ones(1,q);
Pfkxy = (1/(q-1))*((Ekfx)*(Ekfy)');
Pfky = (1/(q-1))*((Ekfy)*(Ekfy)');
K = Pfkxy/(Pfky + R); %Kalman gain

parfor j = 1:q
    v = mvnrnd(zeros(p,1),R)';
    Xka(:,j) = Xkf(:,j) + K*(Yk + v - Ykf(:,j));
end

avr(:,1) = mean(Xka,2);end
Bibliography/References


[16] M. Central. (2012, August) Okada: Surface deformation due to a finite rectangular source. [On-


