Boston College

The Graduate School of Arts and Sciences

Department of Earth and Environmental Sciences

DETECTION OF INDUCED SEISMICITY DUE TO OIL AND GAS EXTRACTION IN THE NORTHERN GULF OF MEXICO, USA

a thesis

by

OLUWASEUN IDOWU FADUGBA

Submitted in partial fulfilment of the requirements

for the degree of

Master of Science

August 2015

DETECTION OF INDUCED SEISMICITY DUE TO OIL AND GAS EXTRACTION IN THE NORTHERN GULF OF MEXICO, USA

OLUWASEUN IDOWU FADUGBA

THESIS ADVISOR: PROFESSOR JOHN E. EBEL

ABSTRACT

Drilling operations and extraction of oil and gas (O&G) may lead to subsurface slumping or compression of sediments due to reduced vertical principal stress which may lead to small earthquakes at the drilling site. O&G extraction is common in the northern Gulf of Mexico (NGM) and only thirty-five earthquakes of magnitudes between 2.3 and 6.0 have been recorded in the area from 1974 to the present. The purpose of this research is to detect more earthquakes using stacks of seismic data from the Transportable USArray (TA) from 2011 to 2013, and determine the spatiotemporal relationship between the detected earthquakes and O&G extraction. Five new small offshore earthquakes, that may be associated with the offshore O&G production, have been detected in the data. Spatial correlation of the epicenters with offshore drilling sites shows that the earthquakes may be due to the O&G extraction.

ACKNOWLEDGEMENT

I want to thank the Almighty God for the gift of life, favor, grace, wisdom, mercy and for the good health and sound mind throughout the course of my program.

I want to appreciate the Department of Earth and Environmental Sciences and all the faculty and staff for giving me the opportunity to study for my master's program here at Boston College on full scholarship with a teaching assistantship and for funding my research during the summers of 2014 and 2015.

I appreciate the efforts of my mommy, Mrs. Victoria Fadugba, my siblings, my pastors, Pastors Tunde and Tinu Olorunwunmi, for their love and support both morally and spiritually.

I will like to thank:

- Professor John Ebel for supervising this research. The invaluable support, guidance, criticisms and advice towards the success of this thesis and my program are innumerable. Thank you so much!!!
- Professor Jeremy Shakun and Professor Seth Kruckenberg for their important and helpful comments in the review of this thesis.
- Oluwaseyi and Olufemi Bolarinwa for always been there for me. Both of you are part of my success story. More grace!!!
- Bernice Siegel, Amy and Jose Luis Sicairos for their love, parental care and for hosting me throughout my program in Boston. Gracias!!!
- Barry Schaudt and Wei Qui for computer programming supports and help with the Boston College Linux Cluster to store and process my data.

- Caleb Lucy for being a true friend. I can’t thank you enough Caleb!!! You are part of my story in Boston.
- Justin Starr, Marilyn Bibeau, Vanessa Napoli, Stephen Hilfilker (my twin brother from another country: same year, month and day of birth; same program; same year of admission; same supervisor; living in the same building and same position in the Boston College GIS Contest – Graduate Level Entry) and all my colleagues for their timing help and cooperation.
- Jesse Brownstein for his help with proofreading my thesis and the wonderful, revealing and important discussions on my thesis. I really appreciate your help, Jesse!

TABLE OF CONTENT

List of figures

List of Tables

CHAPTER ONE: INTRODUCTION

1.1 Introduction ……………………………………………………………………...…1

CHAPTER TWO: GEOLOGIC SETTING AND STUDY METHODOLOGY

2.1 General Geologic Setting in the NGM …………………………………………………6

2.2 Overview of Data Preparation and the Event Extraction Algorithm …………..…7

2.2.1 Narrow Bandpass Filtering …………………………………...………………11

2.2.2 Normal Moveout (NMO) Correction ……………………………………………14

2.2.3 Prestack Automated Separation of the P and S waves …………………………. 20

2.2.4 Hilbert transform……………..…………………………………………………...20

2.2.5 RMS Automatic Gain Control (AGC) ……………………………………..23

2.2.6 Weighted Stacking Technique ………………………………………………..27

CHAPTER THREE: RESULTS

3.1 Introduction …………………...…………………………………………….…...31

3.2 Absolute Location, Origin Time and Depth of the Detected Earthquakes ……...34

3.3 Magnitudes (Mlg) Determination ……………………………………………..47

CHAPTER FOUR: DISCUSSION

4.1 Spatial Correlation of the Detected Earthquakes and Oil and Gas Extraction ….52

4.2 Missing Earthquakes and the Gutenberg-Richter Law ………………………...62

4.3 The Poroelastic Model ……………………………………………………..62

CHAPTER FIVE: CONCLUSION

Conclusion ………………………………………………………………………67

Reference ...…………………………………………………………………….. 69

Appendix A Transportable Array Stations ………………...………………………… 73

Appendix B Event Extraction Algorithm ……………………………………………… 78

Appendix C Algorithm for plotting data points on Google Map …………...………. 101

List of Figures

Figure Page Number

1 Distribution of earthquakes in Lacq field, France …..…………………..……..3

2 Location of 35 previously recorded earthquakes with magnitudes between 2.3 and 6.0 in the Gulf of Mexico from 1974 to the 2014 …….…………..………..….…4

3 Map of the Transportable Array used in this research to study the earthquakes in the NGM…………..…………………………………………………………8

4 Seismic signal processing steps used in the event extraction algorithm ...……..… 9

5 Relationship between signal and background noise………………………….... 10

6 Application of narrow bandpass filter ….……………………………………..…13

7 Event extraction algorithm using (a) 1-10 Hz and (b) 1-2 Hz bandpass filter for the November 4, 2012 event in the time window 19:16-19:20…………….…..…..15

8 Seismic waveforms detected at different TA stations with increasing epicentral distance for the Mississippi Nov 10, 2012 earthquake.………………...………..16

9 Linear regression of the P- and S- wave travel times for the Mississippi 2012 and Gulf of Mexico 2013 earthquakes to determine the apparent velocity in the NGM………………………………………………………………………..18

10 Vertical (BHZ), east-west (BHE) and north-south (BHN) components of the seismic waveforms……………………………………………………………......19

11 Automatic separation of the 4-minute data window into both P and S waves. 20

12 Envelope function of seismic waveforms using Hilbert Transform……..…..…..22

13 Decrease in amplitude with distance due to absorption and spherical divergence24

14 Gain function derived from the Root Mean Square Automatic Gain Control to boost the amplitudes at farther stations……….……………………………….. 26

15 Effect of the RMS AGC correction on the TA waveforms of the Mississippi November 10, 2012 earthquake……………………………………………….... 28

16 Weighted stacking of the background noise (top panel), P wave (middle panel) and S wave (bottom panel) using (a) the envelope functions and (b) without envelope function of the filtered waveforms……….......…………………….29

17 Plot of waveforms at different stations with distance away from Gulf showing the types of events detected between 2011 and 2013 in the NGM……………....32

18 Newly detected earthquakes in the NGM showing the waveforms at all the TA stations with the same moveouts as the previously detected earthquakes in the NGM………………………………………………………………………..33

19 Velocity profile in the NGM with the line of the profile extending from onshore Louisiana to the deep offshore NGM……………………………………..…35

20 Epicenters and their uncertainties of the August 20, 2012 earthquake using six different velocity models for the absolute locations…….…………………….40

21 Epicenters and their uncertainties of the August 26, 2012 earthquake using six different velocity models…………………..……………………………………..41

22 Epicenters and their uncertainties of the September 3, 2012 earthquake using six different velocity models……….……………………………………………..…42

23 Epicenters and their uncertainties of the October 4, 2012 earthquake using six different velocity models…………………..…………………………………..…43

24 Epicenters and their uncertainties of the November 10, 2013 earthquake using six different velocity models…………………………………………………………...44

25 Epicenters and their uncertainties of the December 28, 2012 earthquake using six different velocity models…………………………………………………………...45

26 Epicenters and their uncertainties of the March 11, 2013 earthquake using six different velocity models……………...…………………………………………....46

27 Determination of X and n for the NGM to estimate the magnitude of the detected earthquakes……………………………………………………………………...…48

28 Effect of geology on propagation of P and S wave arrival times and the peak amplitudes which in turn affect the determination of magnitudes……..………...50

29 Effect of station bias on magnitude determination………………..…….………51

30 Location of oil and gas wells in Alabama showing the different types of production with different symbols………..……………………………………...53

31 Locations of gas wells in the Mobile Bay…………………..…………………....54

32 Spatial correlation of drilled wells around the Mobile Bay area with August 20, 2012 epicenter………………………………………...……………………….58

33 Spatial correlation of drilled wells around the Mobile Bay area with August 26, 2012 epicenter…………………………...…………………………………….....59

34 Spatial correlation of drilled wells around the Mobile Bay area with September 3, 2012 epicenter…………………………...……………………………………..…60

35 Spatial correlation of drilled wells around the Mobile Bay area with November 10, 2012 epicenter ………………………………...……………...……........61

36 The plot of the logarithm of the number of earthquakes above certain magnitudes of earthquakes in the NGM from 1974 to 2013 against magnitude……………..…63

37 A schematic cross section of the poroelastic model of Segall (1989) showing the surface deformation, faulting, and simulated focal mechanism associated with oil and gas extraction…………………………………………………………....65

List of Tables

Table Page Number

1 Seismic velocity models used for the determination of absolute locations of the detected earthquakes in the NGM……………………………..………………36

2 68% certainty in earthquake determinations using different velocity models…...38

3 Classification of detected earthquakes and their locations, depths, magnitudes and the uncertainties……………………….……………..…….…………….....39

4 Drilled wells around the Mobile Bay area..55

5 Number of producing wells within 10 km of the epicenter of the earthquakes.... 57

CHAPTER ONE: INTRODUCTION

1.1 Introduction

Most of the earthquakes that have been recorded in the world to date are due to tectonic forces within the earth's crust and mantle. Tectonic forces lead to the formation of faults and release stored potential elastic energy in the rock as seismic waves. Earthquakes are also caused by other phenomena such as landslides and the impact of meteorites on earth. However, many earthquakes have been discovered to be related to or induced by anthropogenic activities, e.g., fluid injection into wells (Segall, 1985, and the references therein; Keranen et al., 2013, etc.). All induced earthquakes are understood to result from increasing pore fluid pressure which reduces the effective mean stress, therefore allowing failure on pre-existing faults at lower shear stress due to the reduction in the effective normal stress across grain boundaries. Surprisingly, some earthquakes are also attributed to fluid extraction. This phenomenon is difficult to conceptualize based on pore-pressure considerations because extraction of fluid should 'delubricate' the faults and act to 'lock' them, thus inhibiting slipping and rock failure as earthquakes (Segall, 1985; Segall et al., 1994, Maury et al., 1992, Baronova et al., 1999 and Bardainne et al., 2008). Induced earthquakes in areas of fluid extraction could be caused by normal faulting that result due to the sudden collapse of voids in the earth after large volumes of fluid have been extracted (Segall, 1985; Segall et al., 1994).

Many earthquakes have been detected in and near oil and gas fields around the world, e.g., the Wilmington oil field in California (Yerkes and Castle, 1970), the Lacq Gas field in France (Segall, 1985; Segall, 1992; Segall et al., 1994; Maury et al., 1992), and the Strachan field in Canada (Baronova et al., 1999). About 90% of the hypocenters of the earthquakes in the Lacq field are located within the oil and gas production area (Maury et al., 1992 in Bardainne et al., 2008; Fig. 1a). The earthquakes in the Lacq field also occurred after the start of production from the field, lending support to the idea that the earthquakes were induced (Segall et al., 1994). Furthermore, these earthquakes have been observed to occur at shallower depth compared to the ones related to the tectonic activity along the Pyrenean fault, which is approximately 25 km from the Lacq field, and so the events are most likely related to the oil and gas extraction in the Lacq field (Segall et al., 1994, Fig. 1b and 1c).

The Northern Gulf of Mexico (hereafter referred to as the NGM) is an oil and gas producing region that is relatively seismically inactive compared to the major plate boundaries seismic zones, e.g., California, Japan, etc., owing to the number of reported earthquakes in the NGM from 1974 to 2013 by the United States Geological Survey (USGS) and the Incorporated Research Institutions for Seismology (IRIS). Thirty-five (35) earthquakes with magnitudes ranging between 2.3 and 6.0 have been recorded in the NGM, primarily in the offshore region, though a few are located onshore (IRIS, 2013, Fig. 2). These earthquakes may be related to the enormous production of oil and gas in the Gulf of Mexico (Texas, Louisiana, Mississippi, Alabama and Florida). For instance, 633 million barrels, 831 million barrels and 1043 million barrels were extracted in the NGM in 2011, 2012 and 2013, respectively (U.S. Energy Information Administration, 2014).

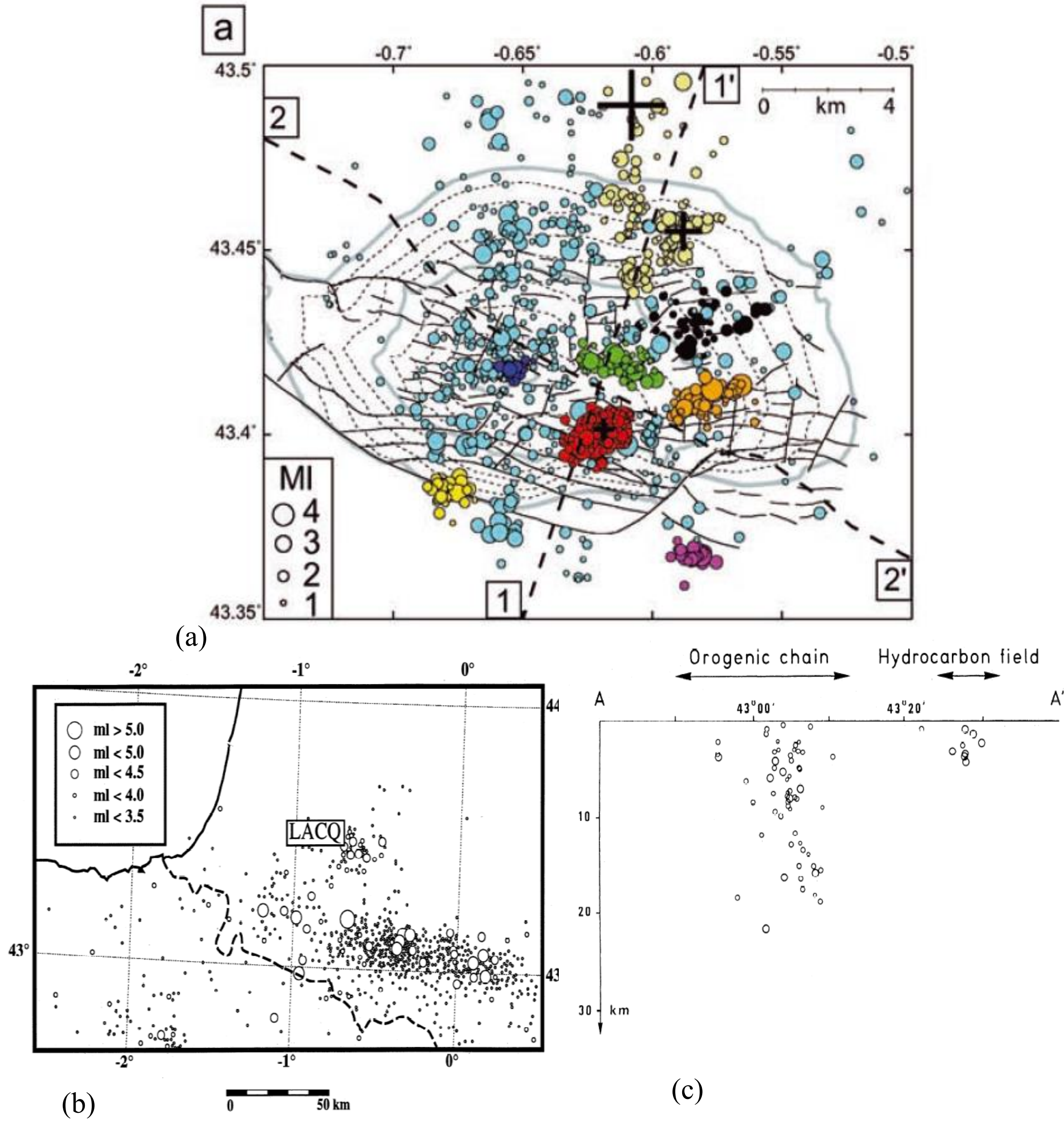


Figure 1: Distribution of earthquakes in Lacq field, France. (a) The different colors show the various earthquakes clusters within the field (Maury et al., 1992 in Bardainne et al., 2008). The extent of the field is shown in gray and the gas pressure contours are represented by dashed lines. It is seen that most of the earthquakes occurred within the field. Some local faults are identified with solid black lines. (b) Earthquake epicenters in the Pyrenees from 1962-1990 for events having magnitudes greater than 3. (c) North-south vertical cross section of these earthquakes from October to December 1982. It can be observed that the earthquakes in the Lacq field are separated spatially from those associated with tectonism along the Pyrenean fault and occurred at shower depths (Segall et al., 1994).

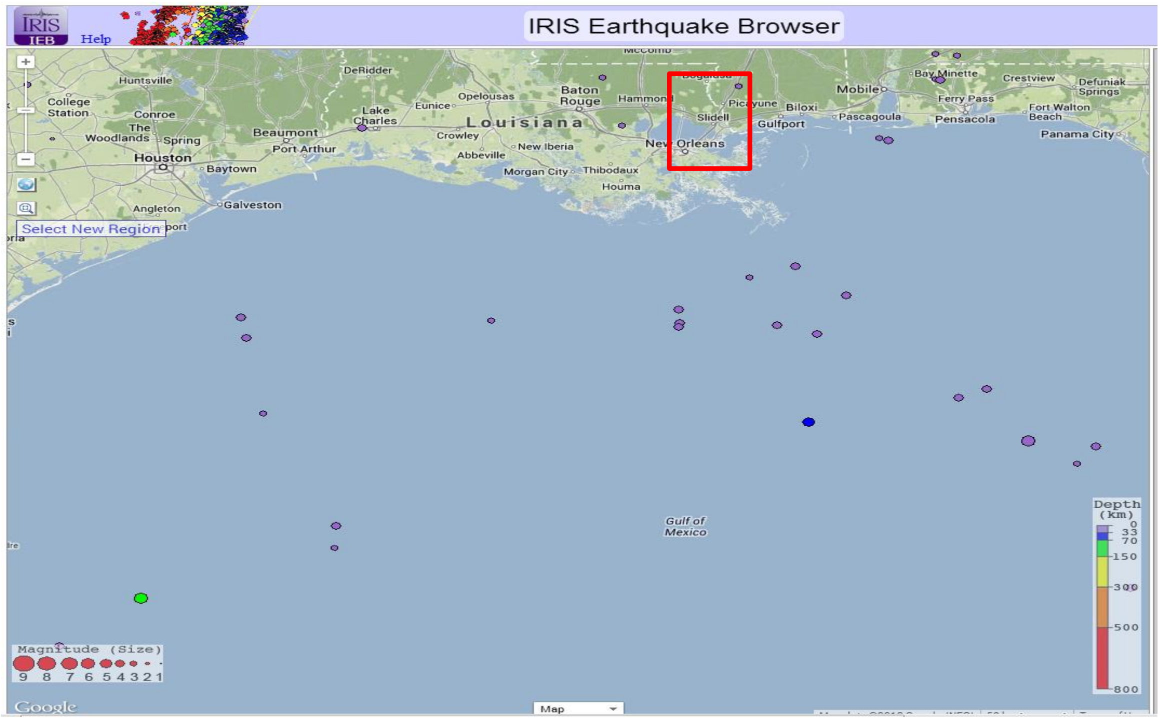


Figure 2: Location of 35 previously recorded earthquakes with magnitudes between 2.3 and 6.0 in the Gulf of Mexico from 1974 to the 2014 (IRIS, 2013). The red rectangle shows the Mobile Bay area and the two earthquakes detected within that gas field on February 18th, 2011 and November 10th, 2012.

The purpose of this research is to search for undetected small offshore earthquakes in the NGM using EarthScope Transportable Array (TA) data with a new earthquake detection technique that makes use of an increased signal-to-noise ratio (SNR) of small regional earthquakes by stacking seismic waveforms from different seismometers that were operated during the period of investigation. The research also investigates the spatial correlation of the detected earthquakes with offshore drilling sites in the NGM, and makes an assessment of whether or not recent offshore production is correlated with induced seismicity.

CHAPTER TWO: GEOLOGIC SETTING AND STUDY METHODOLOGY

2.1 General Geologic Setting in the NGM

The Gulf of Mexico is a tectonic basin formed by the opening of North and South America in Callodian/Oxfordian time (ca. 160 Ma) (Meschede and Frisch, 1998). A broad 15 km-thick wedge of Mesozoic-Cenozoic deposits accumulated in the basin, and deposition is still ongoing. The supply of sediments, mainly delivered by the Mississippi River, into the NGM generally exceeds the subsidence rate, causing the sediments to prograde seaward from the shoreline (Holt et al., 1983). The NGM is a passive continental margin that is morphologically comprised of a continental shelf, slope, rise, and abyssal plain and extends from Texas to Florida. The sediments are predominantly coastal sands underlain by Cretaceous carbonate deposits (Ewing et al., 1955).

Salt diapirs, growth faults and shale uplifts are the major structural features in the NGM (Holt et al., 1983). The dominant structures in the offshore sedimentary formations from Louisiana to Texas are different from that west of Florida. Salt domes, growth faults and differential subsidence are abundant in Louisiana and beneath the continental slope east of Texas due to the overloading of the continental shelf and differential compaction of the sediments, whereas karst and unstable slopes are the only geologic features on the west Florida shelf (Holt et al., 1983). The total depth of the carbonate strata west of Florida is approximately 4.57 km and the karst is predominantly porous Eocene limestone and contains many sinkholes formed either by solution of surface limestone or by the collapse of underlying solution caverns (Holt et al., 1983).

2.2 Overview of Data Preparation and the Event Extraction Algorithm

The waveform data from the Transportable Array (TA) stations in the Gulf Coast states between 2011 and 2013 were used in this research. The waveform data from 52, 43 and 44 TA stations were used for the detection of earthquakes for 2011, 2012 and 2013 respectively (Fig. 3, Appendix A). The waveform data were retrieved from the IRIS Data Management Center in SEED format and were subsequently converted to SAC format using the Jrdseed program for processing (IRIS, 2013). The TA stations are temporary seismic stations deployed on the continental US to measure ground motions for studying subsurface tectonic activities and structures (IRIS, 2013). The TA stations sample at 40 Hz in the vertical (BHZ), North-South (BHN) and East-West (BHE) broadband components. The dense configuration and proximity to the NGM make the TA stations effective at detecting the expected small shallow offshore earthquakes.

The step-by-step processing sequence for the event detection system is summarized in a flowchart (Fig. 4, Appendix B). The input into the algorithm is the waveform from the TA stations in SAC format. The algorithm for extracting station waveforms used data for a 4-minute window with a 2-minute overlap with the prior 4-minute interval to ensure the detection of earthquakes that may have their origin time near the boundary between two adjacent processing windows (Fig. 5). A narrow bandpass filter of 1-2 Hz was applied to the data in the 4-minute seismogram window to remove background noise and high-frequency signals and noises.

A prestack RMS Automatic Gain control (AGC) was applied to boost the amplitudes of the signals at farther stations from the northern NGM to compensate for the

Transportable Array Stations

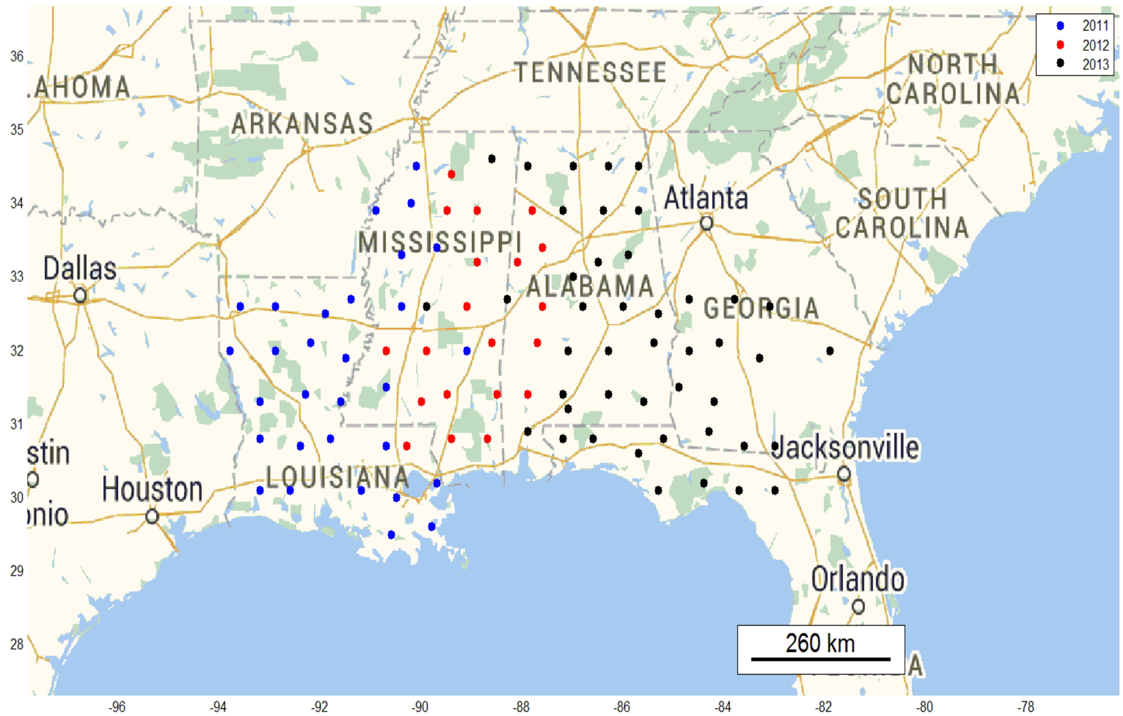


Figure 3: Map of the Transportable Array used in this research to study the earthquakes in the NGM. The blue circles represent the TA stations to study in 2011, red circles represent 2012, and black circles represents those used for the year 2013. There is an overlap between the circles.

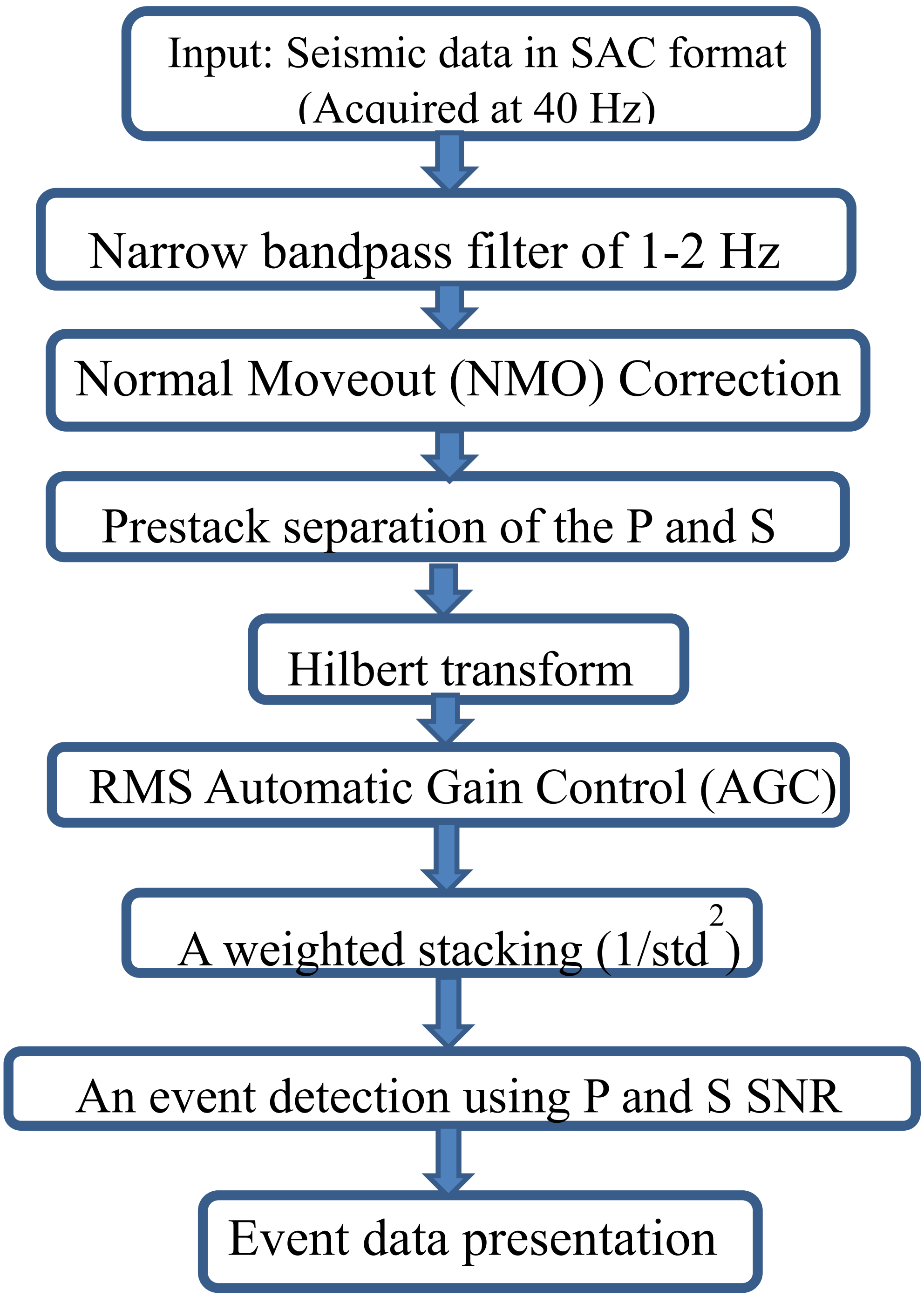


Figure 4: Seismic signal processing steps used in the event extraction algorithm.

Station AL TA-349A BHZ (10-04-2012)

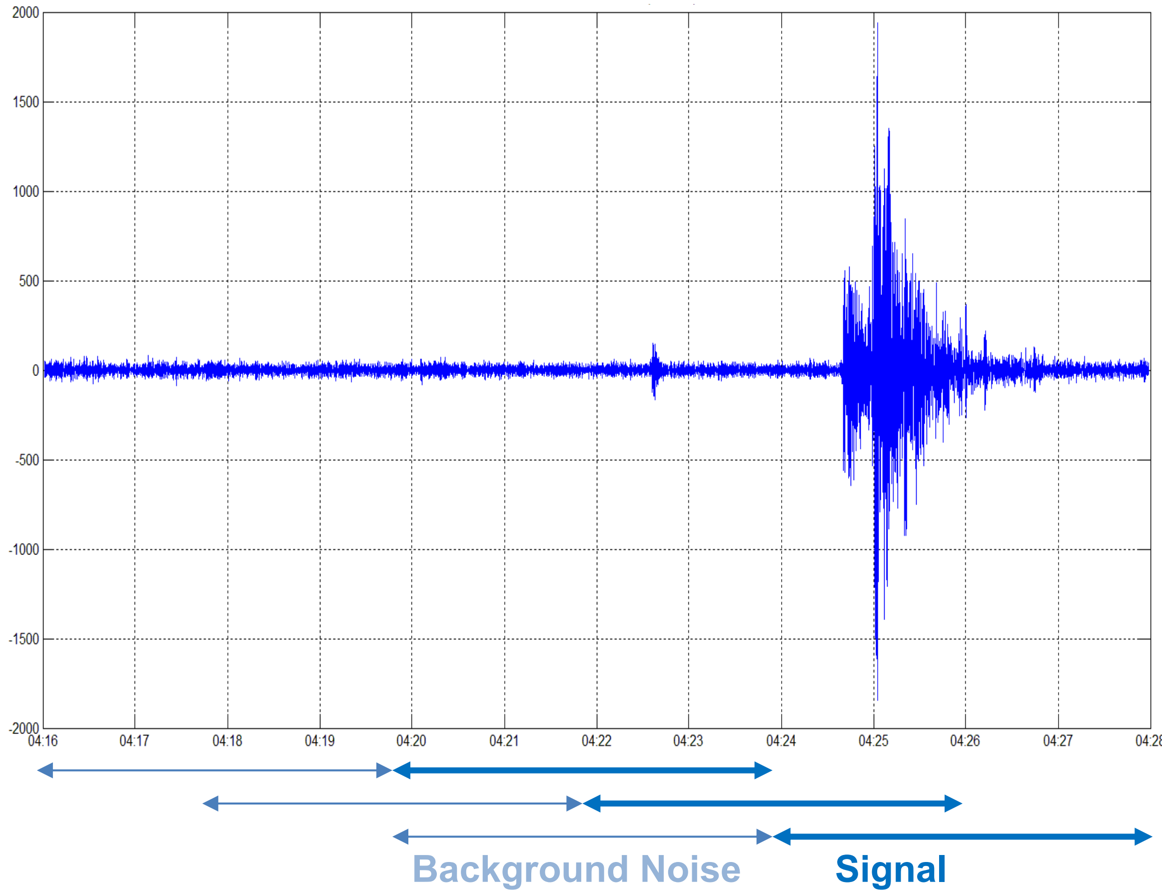


Figure 5: Relationship between signal and background noise. The processing proceeds with 4-minute intervals with 2-minute overlap. The results of processing the previous 4-minute window were used as the background noise.

decrease in signal amplitudes from northern NGM sources due to spherical divergence and anelastic attenuation. Previously known earthquakes were analyzed for the proper Normal Moveout (NMO) velocity in order to correct for the moveout of the P- and S-waves across the seismic stations due to the differences in the arrival times as the earthquake energy propagates from the epicenter. A Hilbert transform was used to determine waveform envelopes in order to avoid destructive stacking, and a weighted stacking method was then applied separately to the P and S waves. An event was considered detected when the P- and S- signal to noise ratios (SNRs) are both greater than 10, a high threshold that was set in order to minimize the rate of false detections. The seismic records for all the TA stations used in this analysis were plotted with respect to epicentral distance. The plotting method is effective for the visualization, classification and discrimination of events using a collection of several seismic waveforms from adjacent TA stations. The Linux cluster maintained by the Boston College Research Services was used for the data processing and was accessed remotely from a personal computer, a necessary method due to the enormous volume of data (> 1 Terabyte) and the extensive processing capability that was required.

The various processing sequence steps and their mathematical backgrounds are described in the following subsections.

2.2.1 Narrow Bandpass Filtering

P and S waves are not dispersive, i.e., the earthquake waveform has all frequencies and these frequencies all travel at the same speed. A bandpass linear-phase finite impulse response filters allow the passage of energy at a specified range of frequencies in the signal and remove the energy at other frequencies (Schilling and Harris, 2012). The linear-phase property makes the output waveforms from these filters to be uniformly shifted with no distortion in the shape of the output signal compared to that of the input signal.

A narrow bandpass Butterworth filter of 1 – 2 Hz was applied to each 4-minute window of TA waveform records in order to remove high-frequency background noise and signal, thereby focusing on the lower frequency components of the signal (Fig. 6). A filter of zero transition band is called an ideal filter and cannot be realized in practice (Schilling and Harris, 2012). The smaller the transition band, the more ideal the filter. Increasing the filter order can be used in practice to shorten the transition band as small as possible, although higher filter orders require more processing time and computer capability. A filter order of four was used in this research. A shortcoming of Butterworth filters is the introduction of a phase-shift in the waveforms as opposed to a zero-phase filter which introduces no time shift of the input signal. In order to realize a linear phase filter in the application of a non-linear phase filter, a fourth-order zero-phase filter was actualized by applying a second-order filter twice using the *filtfilt* command in MATLAB. The *filtfilt* command filters a signal, and then the time-reversed output is used as the input for another pass through the filter, after which the result is time-reversed once again. Through this process, the final output is a zero-phase with no distortion in the shape of the signal.

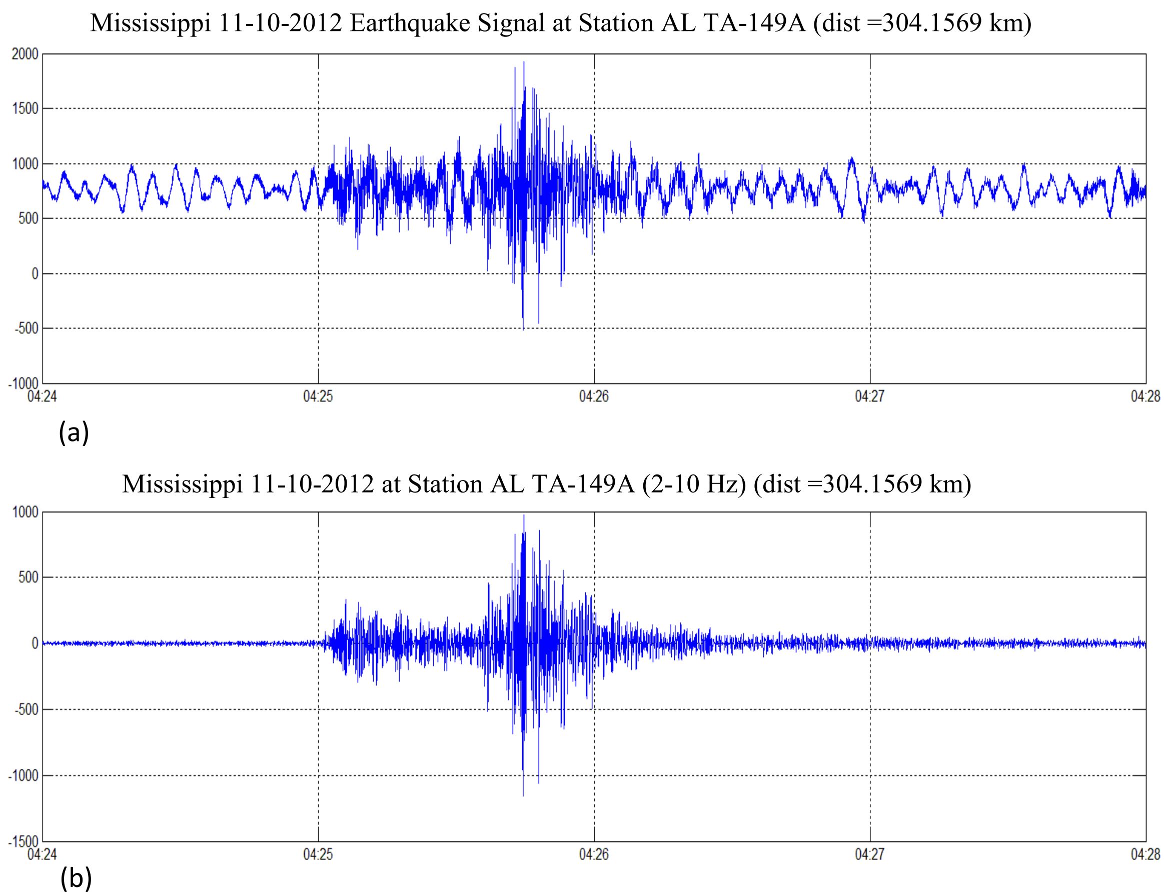


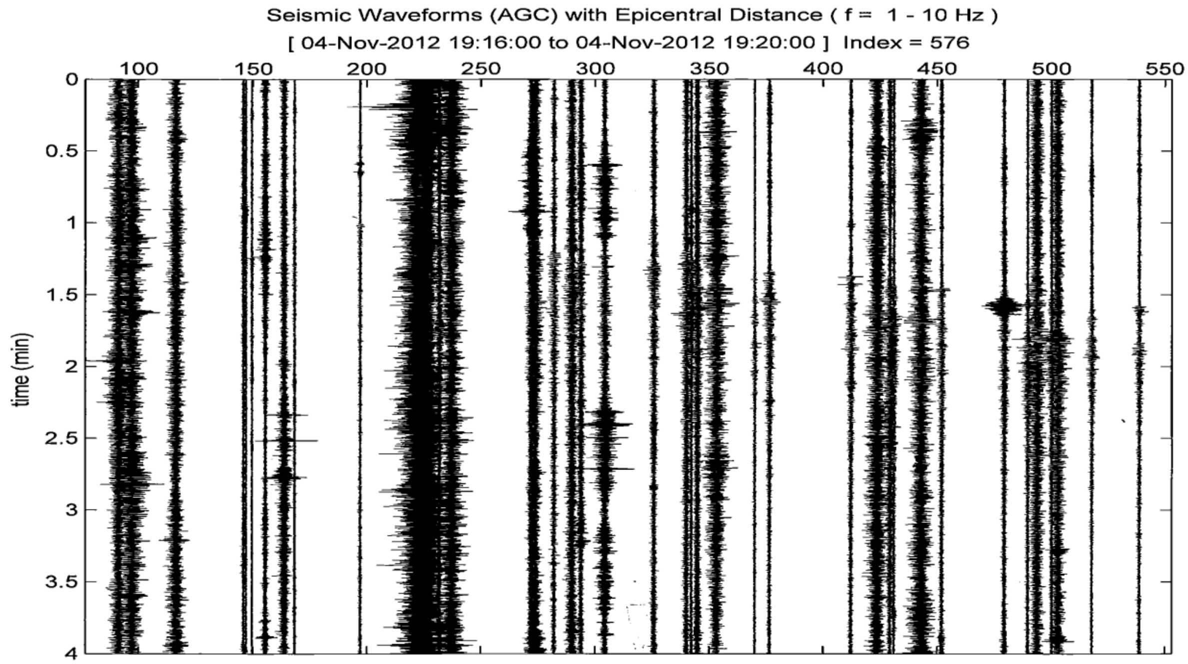
Figure 6: Application of narrow bandpass filter on the (a) raw waveform to give (b) a filtered waveform. The signals from the earthquake are better shown in the filtered waveform.

SNR is a signal processing technique used to compare the energy in a signal (P and S waves) with that of the background noise. The energy of the P and S waves and the background noise were determined by the sum of squares of the amplitudes in the waveforms and its value increases with the square root of the number of the TA stations stacked (Schilling and Harris, 2012). Lamda is the ratio of the standard deviations of the stacked P- and S- waveforms to that of the stack of background noise (Gendron et al., 2000). The previous 4-minute window of data was filtered and used as the background noise. Various bandpass filters were used to filter the known November 10, 2012 earthquake waveform data and their SNR values and lamdas were determined. The various bandpass filters were used in order to determine the filter parameters that gave the highest SNR and lamda and hence best enhance the detection of small shallow earthquakes in the NGM. The narrow bandpass filter of 1-2 Hz gives higher values of signal-to-noise ratio and lamda. Figure 7 shows a comparison of the results when the 1-2 Hz bandpass filter and a wider 1-10 Hz bandpass filter were used to detect the same earthquake.

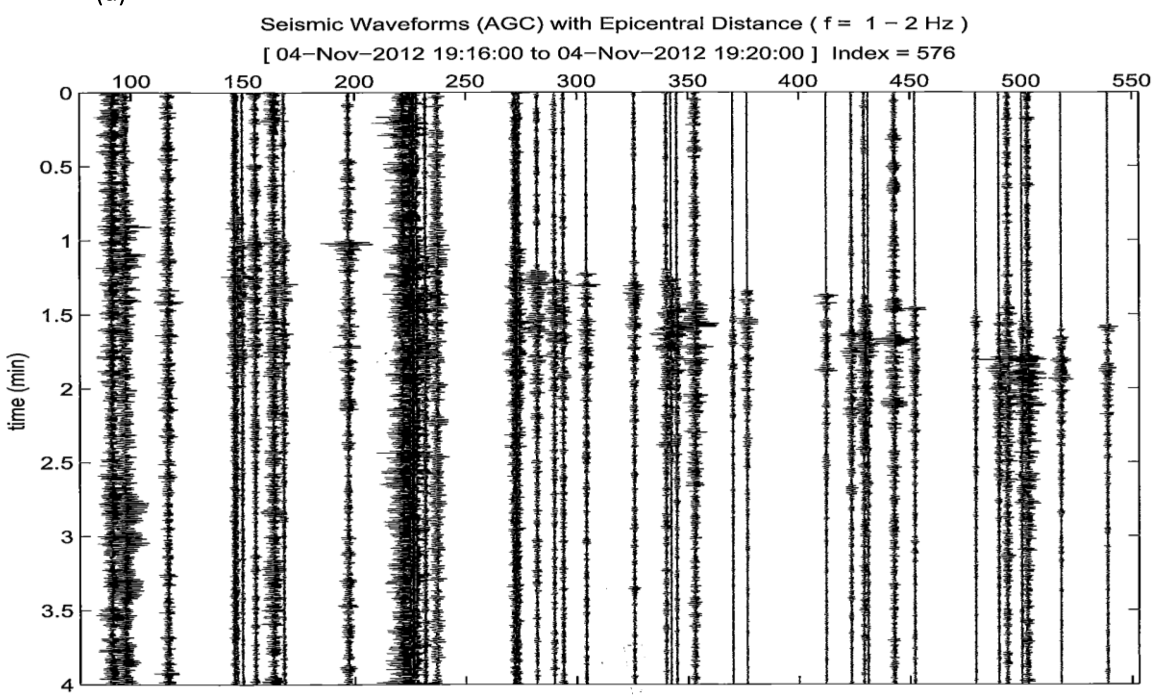
2.2.2 Normal Moveout (NMO) Correction

The propagation of seismic waves from an epicenter across the TA stations creates differences in arrival times for both P and S waves as the waves propagate from station to station (Fig. 8). Because of this effect, the different arrival times need to be corrected to a common time before the filtered waveforms from the various TA stations can be stacked coherently.

Event Extraction Algorithm using 1-10 Hz and 1-2 Hz bandpass filters



(a)



(b)

Figure 7: Event extraction algorithm using (a) 1-10 Hz and (b) 1-2 Hz bandpass filters for the November 4. 2012 event in the time window 19:16-19:20.

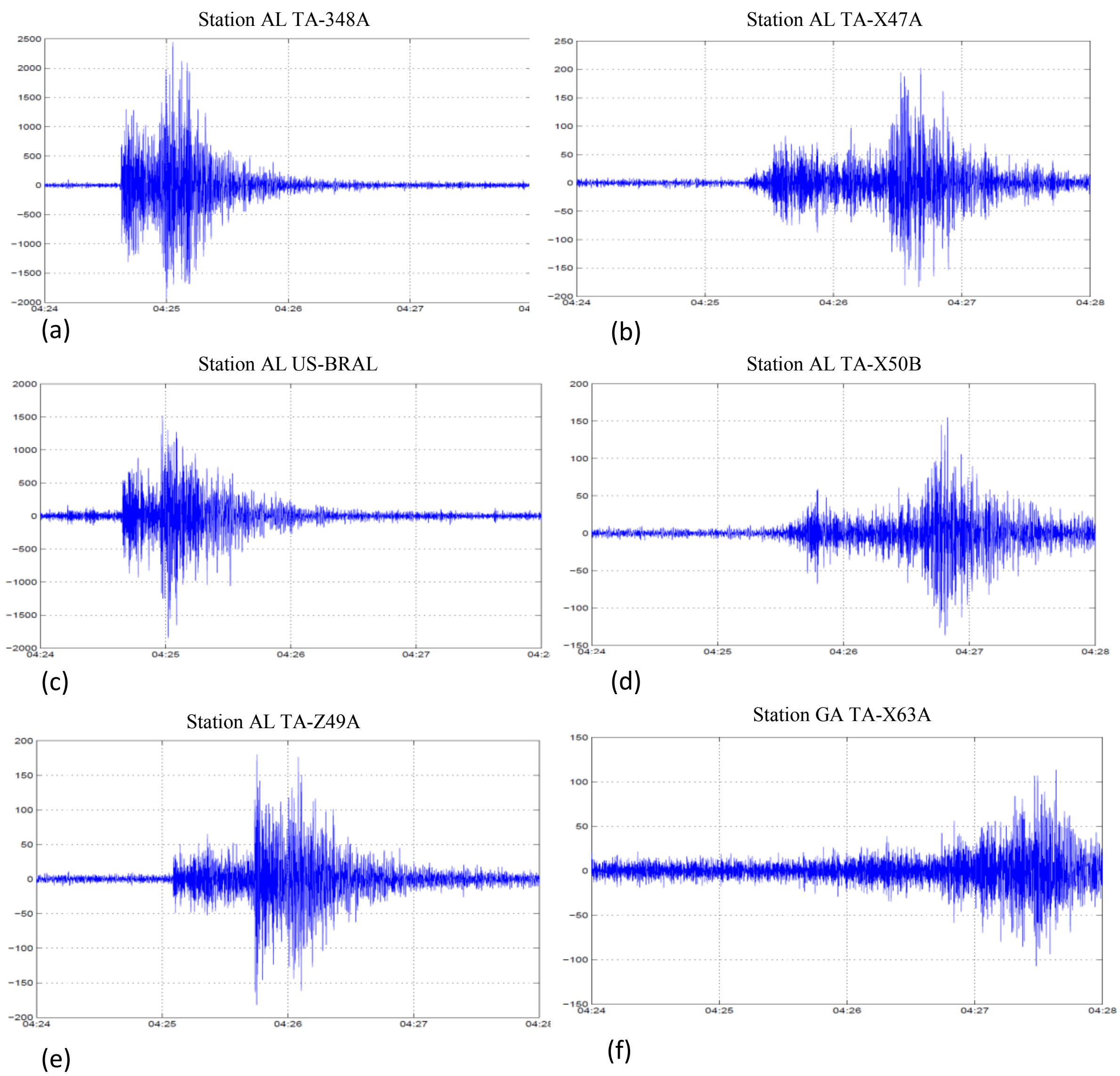


Figure 8: Seismic waveforms detected at different TA stations with increasing epicentral distance for the Mississippi Nov 10, 2012 earthquake. The six waveforms (a-f) have the same horizontal scale. The waveforms show the propagation of the energy and an increase in the S-P arrival time as the wave propagates across and is detected by the TA stations.

Similar to the NMO correction in reflection seismology, the technique applied here to correct the P and S arrival times requires a moveout velocity. P and S wave moveout velocities in the NGM were estimated empirically by plotting the arrival times of the Mississippi November 10, 2012 and the Gulf of Mexico March 11, 2013 earthquakes at the various TA stations against their respective epicentral distances (Fig. 9). The S-wave arrival time can be difficult to pick on the vertical component waveforms, especially at noisy TA stations, and so displaying the waveforms in the north-south and east-west components enhances the picking of the first arrival. Therefore, seismic waveforms in north-south, east-west and vertical components of ground motion at the TA stations were retrieved using the *Iris.fetch.m* MATLAB function in order to improve the S wave arrival time picks (Fig. 10). The S wave arrival times are easier to pick on the east-west and north-south components than the vertical component.

The P and S wave traveltimes for both earthquakes show a linear relationship with epicentral distance with correlation coefficients of 92.54% and 90.41%, respectively (equation 2.7a and 2.7b). The moveout velocities were determined using the inverse slope of the linear trend resulting in 7.669 km/s and 4.439 km/s for the P and S waves velocities, respectively. Due to the high regression coefficients between the arrival times and epicentral distances, the estimated moveout velocities were representative of the P and S wave velocities and are thus suitable for the NMO corrections.

$$\Delta t_{PNMO} = 0.1304x \text{ (P-wave)} \qquad (2.7a)$$

$$\Delta t_{SNMO} = 0.2253x \text{ (S-wave)} \qquad (2.7b)$$

where Δt_{PNMO} and Δt_{SNMO} represent the NMO correction for the P and S waves, respectively, and x represents the epicentral distance.

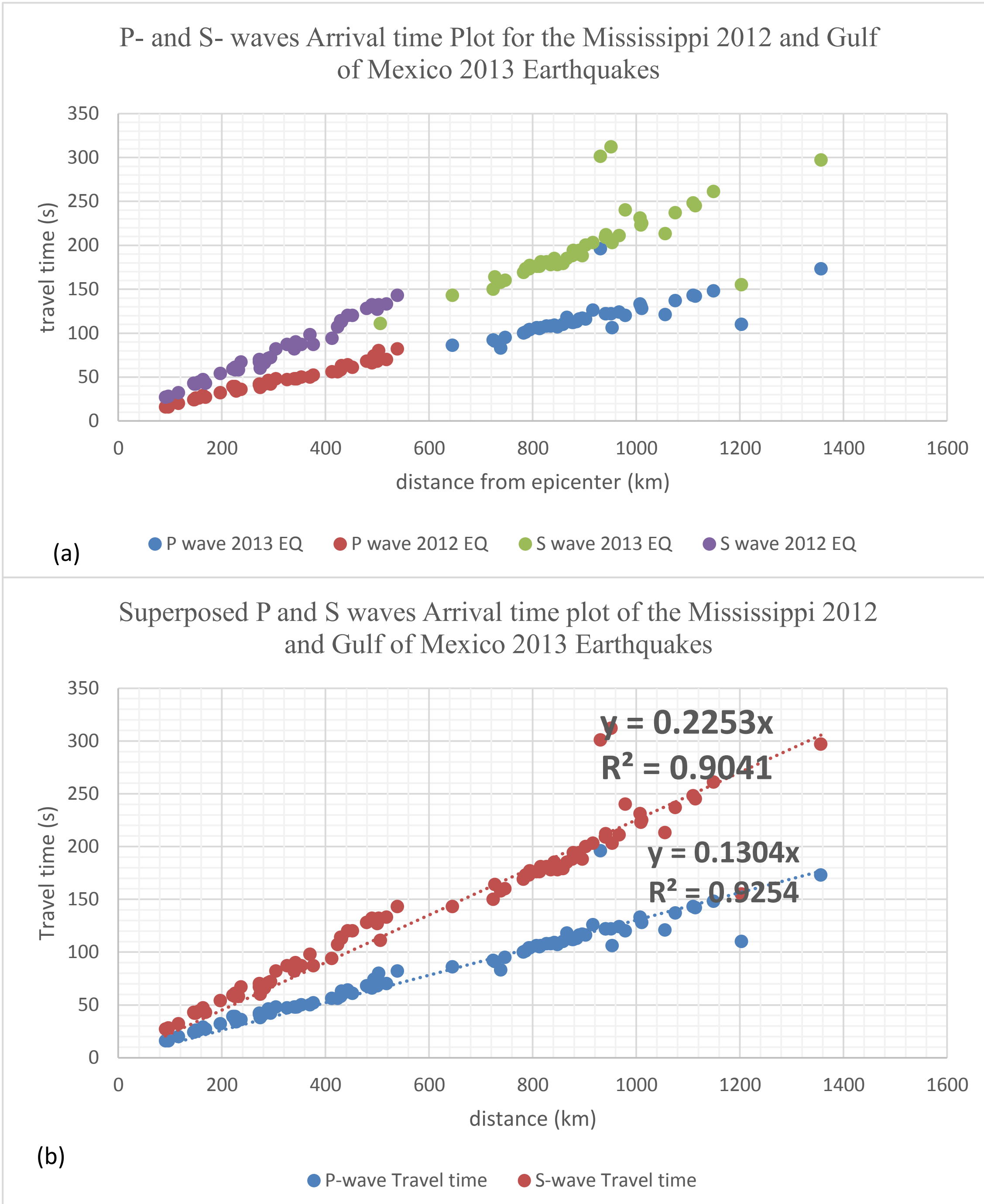


Figure 9: Linear regression of the P- and S- wave travel times for the Mississippi 2012 and Gulf of Mexico 2013 earthquakes to determine the apparent velocity in the NGM. The apparent velocity is used for the NMO correction before stacking the P and S waveforms. (a) The red and purple circles represent the P and S waves respectively for the Mississippi 2012 earthquakes while the blue and green circles represent the P and S waves respectively for the Gulf of Mexico 2013 earthquakes. (b) The blue and red circles represent the P and S waves respectively for the two earthquakes.

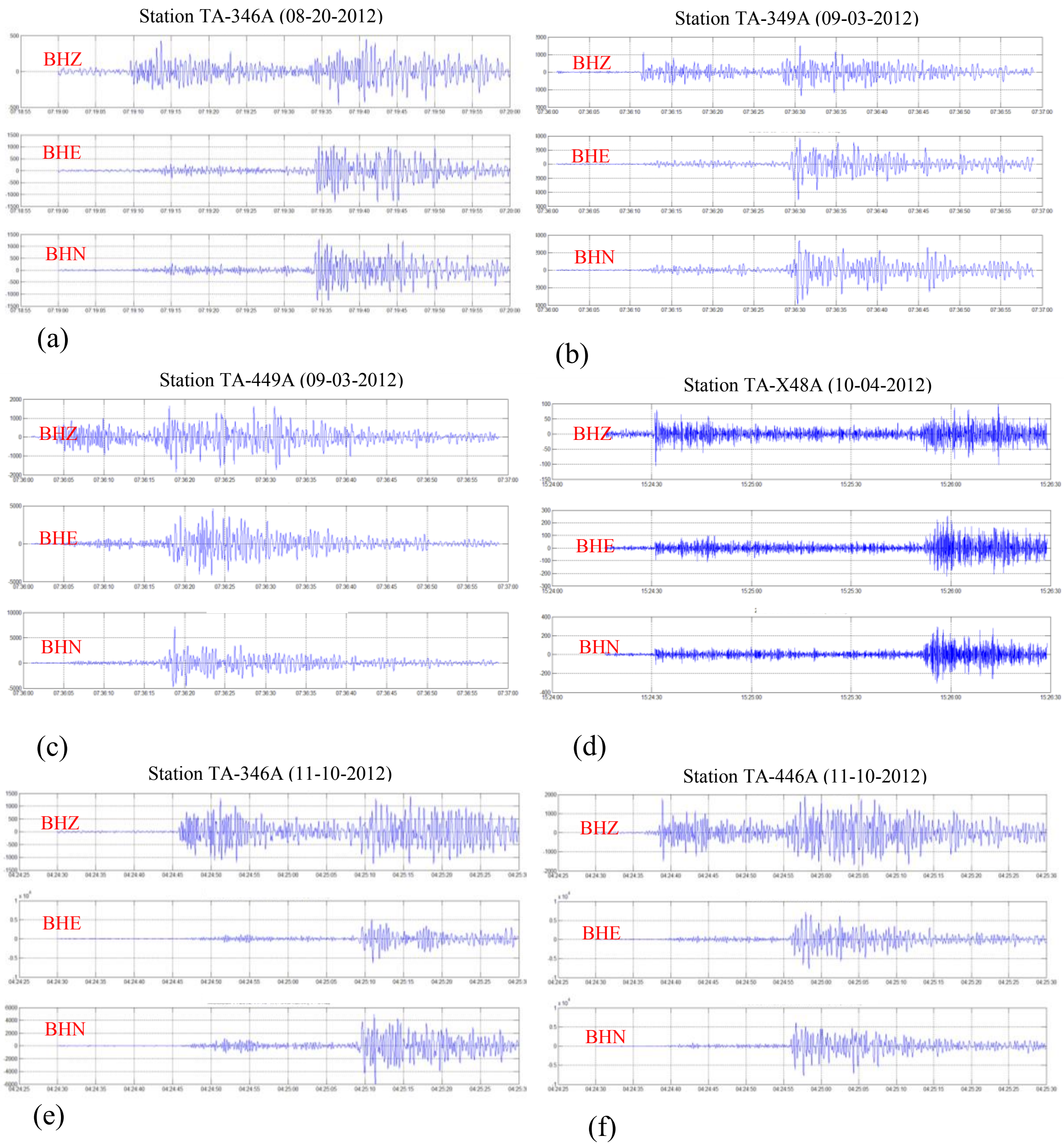


Figure 10: Vertical (BHZ), east-west (BHE) and north-south (BHN) components of the seismic waveforms for the (a) August 20, 2012 (b) and (c) for September 3, 2012 earthquake, (d) for October 4, 2012 and (e) and (f) for the November 10, 2012 earthquake. The S wave arrival times are easier to pick on the east-west and north-south components than on the vertical components.

2.2.3 Prestack Automated Separation of the P and S waves

After the P and S waves have been corrected for the travel-time moveout depending on their moveout velocities, they were separated before the waveforms from the various TA stations were stacked. This separation was necessary to avoid seismic phase stacking interference due to the difference in the S and P wave arrival times as the earthquake energy propagates across the array of stations (Fig 11). Separating the P and S arrival time windows also helps to better identify which sources are from the NGM as opposed to some other source locations. This is because the P and S moveout velocities are different for different seismic source locations and the research interest in this thesis is in the earthquakes from the NGM.

Equations 2.7a and 2.7b were used to separate the P and S waves automatically for each seismic station by predicting the arrival time of the P and S waves from any hypothetical epicenter from the NGM using the moveout equation (equations 2.7a and 2.7b). The P wave window was separated by extracting the waveform data starting at the predicted P wave arrival time and with length equivalent to the difference in the S and P wave arrival times. The S wave window was separated by extracting the waveform data from the predicted S wave arrival time.

2.2.4 Hilbert transform

A Hilbert transform was applied to the P and S waveforms at each seismic station in order to avoid destructive stacking of positive and negative amplitudes in the waveform data (Fig. 12). For example, the stack of two equal but opposite amplitudes

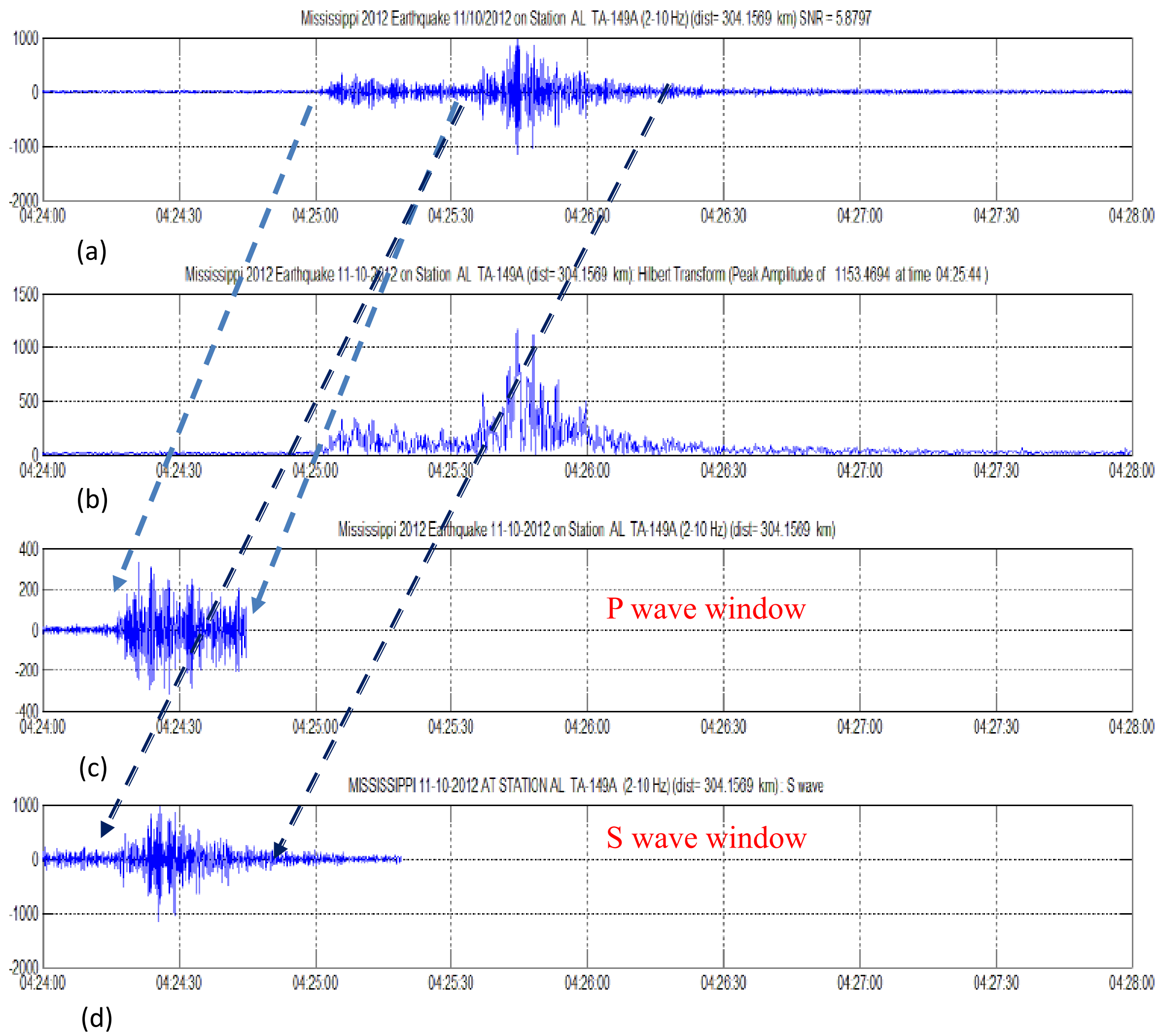


Figure 11: Automatic separation of the 4-minute data window into both P and S waves. (a) 4-minute data window (b) Envelope function of the filtered waveform in (a) using Hilbert Transform. (c) P wave window (d) S wave window.

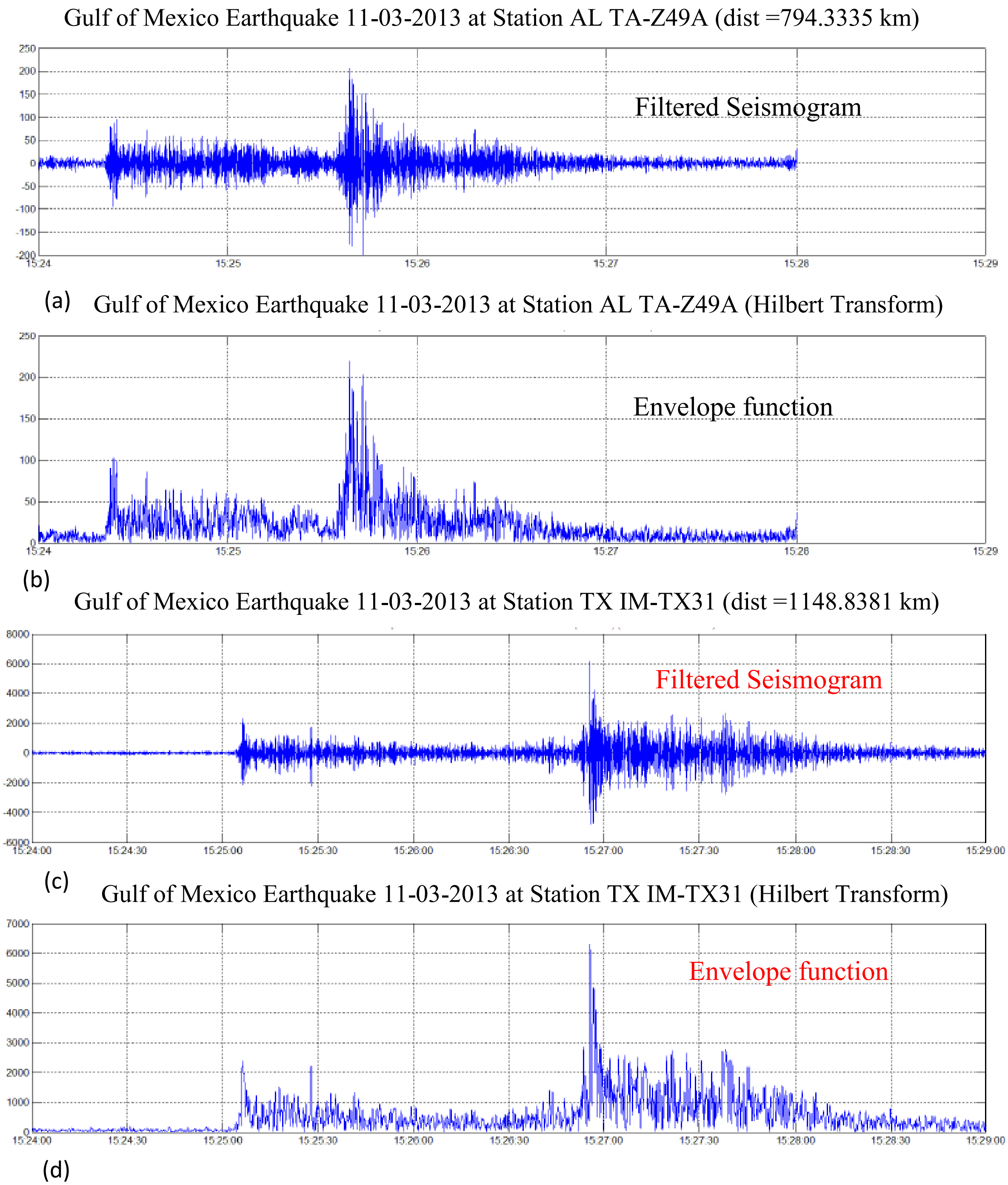


Figure 12: Envelope function of seismic waveforms using Hilbert Transform. (a and c) Filtered seismograms at stations AL TA-Z49A and TX IM-TX31, respectively. (b and d) Envelope function using a Hilbert Transform on (a and c), respectively.

results in a signal with zero amplitude. This destructive stacking is undesirable in the waveform processing since the purpose of the processing is to detect a small shallow earthquake; hence an envelope function using a Hilbert transform was applied. The Hilbert transform is a filter that produces a steady-state output that is in quadrature phase with the input signal (Schilling and Harris, 2012).

The frequency and impulse responses of the Hilbert transform are given as

$$H_d(f) = -j \operatorname{sgn}(f),\ 0 \le |f| \le f_s/2 \quad \text{.................................(2.8 a)}$$

$$h_d(k) = \begin{cases} \frac{1-\cos(k\pi)}{k\pi}, & k \neq 0 \\ 0, & k = 0 \end{cases} \quad \text{.................................(2.8 b)}$$

where the signum or sign function, sgn (f) is given as:

$$sgn(f) = \begin{cases} 1, & f > 0 \\ 0, & f = 0 \\ -1, & f < 0 \end{cases} \qquad \text{............................ (2.9)}$$

2.2.5 RMS Automatic Gain Control (AGC)

As expected, peak amplitudes of waveforms from the Mississippi 2012 earthquake decrease with epicentral distance due to spherical divergence and anelastic attenuation of the signals with increasing distance of propagation (Fig. 13 a and b). The stack of small-amplitudes signals from stations located farther away with high-amplitudes signals from stations nearer to the Gulf reduces the amplitude of the stacked waveform, and hence the signal-to-noise ratio, compared to the stacked waveform when the small-amplitude waveforms were not used or were boosted in amplitude. The presence of many

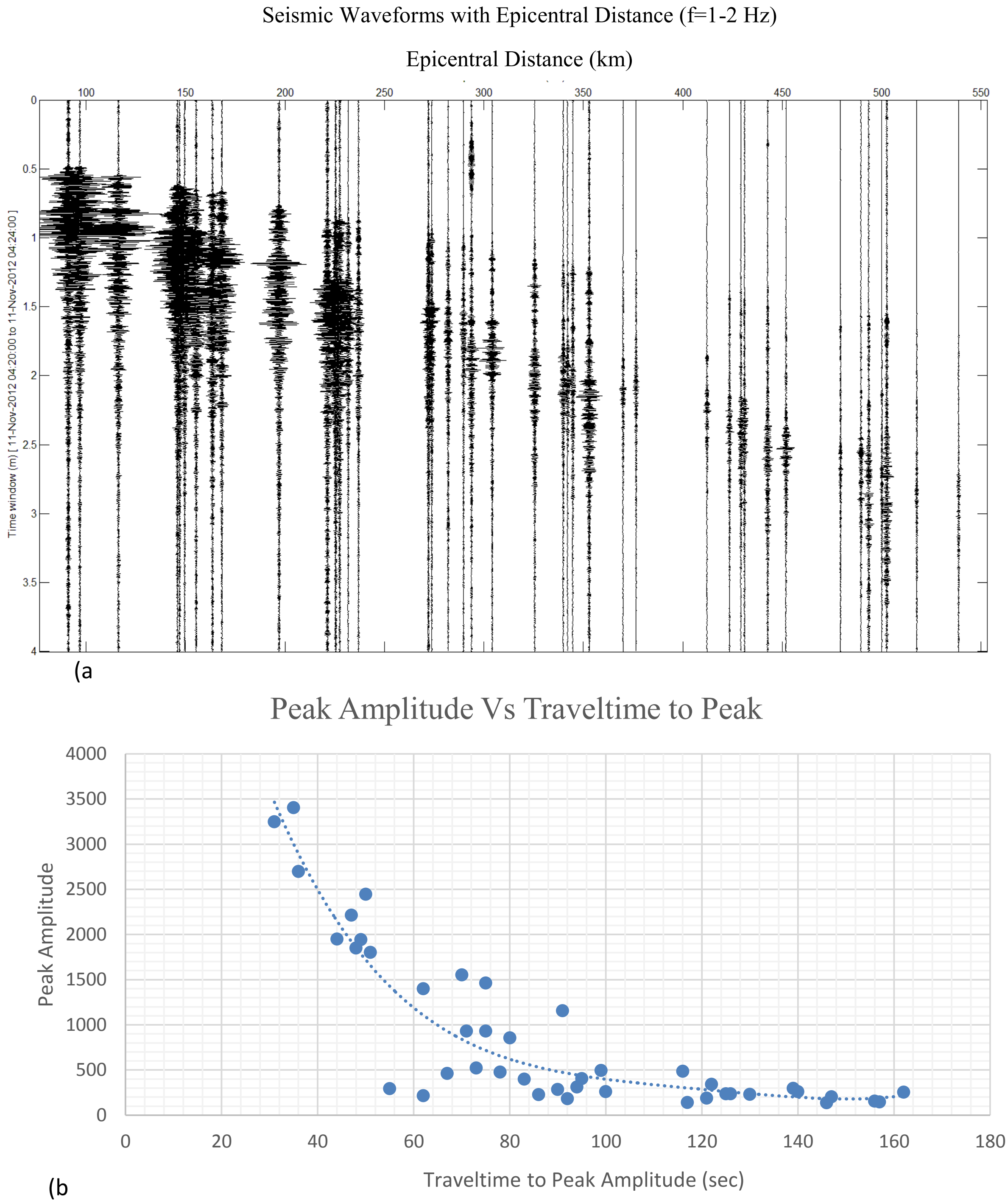


Figure 13: Decrease in amplitude with distance due to absorption and spherical divergence. (a) Plot of various TA seismic waveforms of the November 10, 2012 earthquake against epicentral distance. (b) Plot of peak amplitude of the waveforms against distance. The plot shows that the amplitude decays exponentially with depth.

small-amplitude traces relative to high-amplitude ones reduces the chance of detecting earthquakes within the stacked trace since the detection depends on the value of the signal-to-noise ratio. Furthermore, TA stations nearer to the NGM are often noisy due to the relatively low-velocity unconsolidated sediments at the stations, which increase the amplitudes of the waveforms at those stations. On the other hand, the contributions of the amplitudes of the waveforms at farther stations are paramount for the detection of a small shallow earthquake in the NGM. Hence, an increase of amplitudes of waveforms at farther stations is important in detecting small shallow earthquakes.

A RMS Automatic Gain Control (AGC) was applied to boost the signals at farther stations by multiplying the amplitudes of the waveform at each station by a gain derived from a gain function. The gain function was determined by dividing the TA stations based on their distances from the epicenter of the November 10, 2012 earthquake into 50 km intervals irrespective of their azimuths. The RMS value of the amplitudes of the waveforms in each interval was determined. The gain value at each interval was determined from the ratio of a desired RMS value and the RMS value of the amplitudes of the waveforms within the interval (Yilmax, 2001). The desired RMS value equaled 10,000 counts in this study, due to the wide range of amplitudes of the waveform at proximal compared to distal TA stations from the NGM. The gain value for each interval was plotted at the end of the interval (e.g., 50 km epicentral distance for the first interval). The gain values at each 50 km interval were connected with lines with the assumption of a constant rate of change in the gain as a function of distance between successive gain values (Fig. 14). Thereafter, the gain value at each TA station was extracted from the plot of gain with epicentral distance at their respective epicentral distances. The extracted

RMS Amplitude AGC with Epicentral Distance

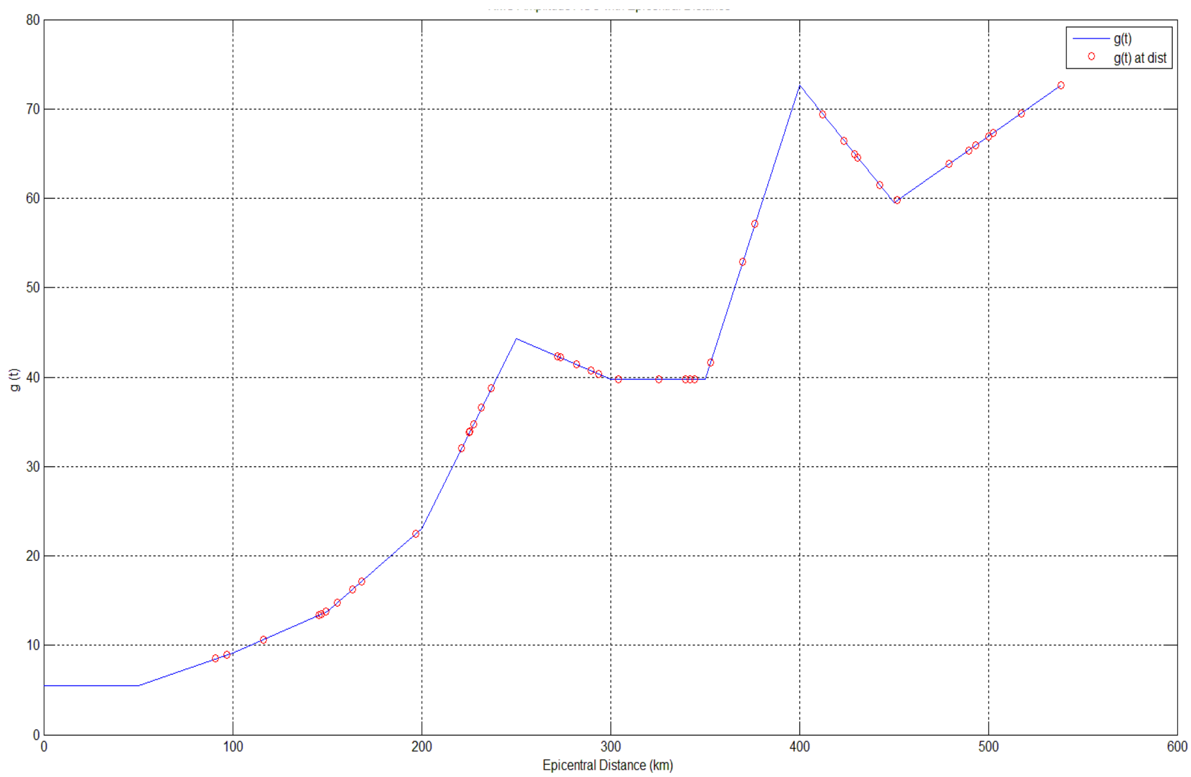


Figure 14: Gain function derived from the Root Mean Square Automatic Gain Control to boost the amplitudes at farther stations. The two peaks are probably as a result of reverberations in subsurface layers. The blue segmented line represents interpolated function derived from the gain value at each station within each 50 km interval. The red circles represent the gain extracted from the gain function for each TA station based on its epicentral distance.

value for each TA station was subsequently multiplied with the amplitudes of the waveform data.

The result of the RMS Automatic Gain Control technique compared to the original TA waveforms shows an increase in the amplitudes of farther stations and a decrease in the amplitudes at nearer TA stations (Fig. 15).

2.2.6 Weighted Stacking Technique

Signal-to-noise ratio can be increased by stacking many waveforms at different TA stations, and an increase in the number of station (N) used increases the signal to noise ratio (SNR) by a factor of square root of N if the noise is random and uncorrelated (Schilling and Harris, 2012). In order to reduce the contribution of the waveforms with low signal-to-noise compared to the waveforms with high signal-to-noise ratio, a weighted stacking based on the standard deviations of the background noise of the waveforms are used. A weighted stacking technique was applied to the envelope function of the AGC- and NMO-corrected P and S waves using an inverse square of the standard deviation of the background noise as the weight of the stack. The background noise is found from the 4 minutes preceding the 4-minute data window studied. However, if an event was detected in the preceding 4-minute data window, the data window between the 4 and 8 minutes before the 4- minute signal data window was used as the background noise. The envelope function was used to avoid destructive stacking due to different polarity of the amplitudes (Fig. 16). The equation used for this process can be expressed as:

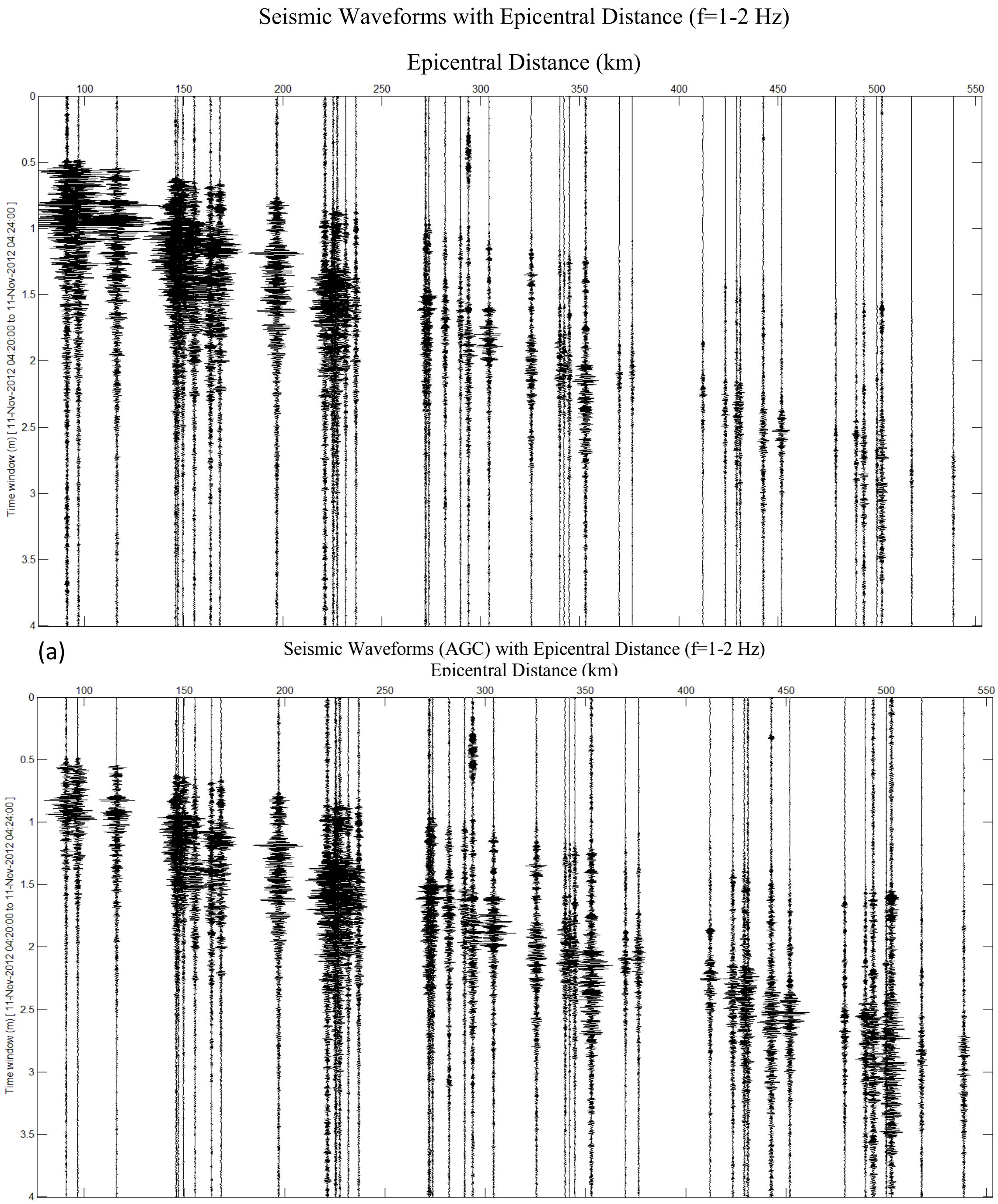
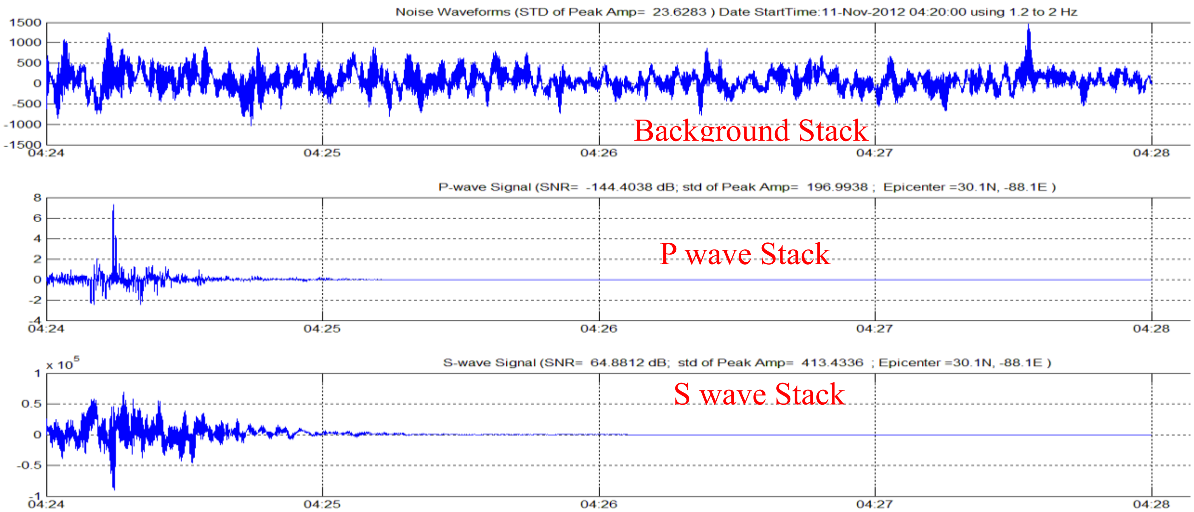
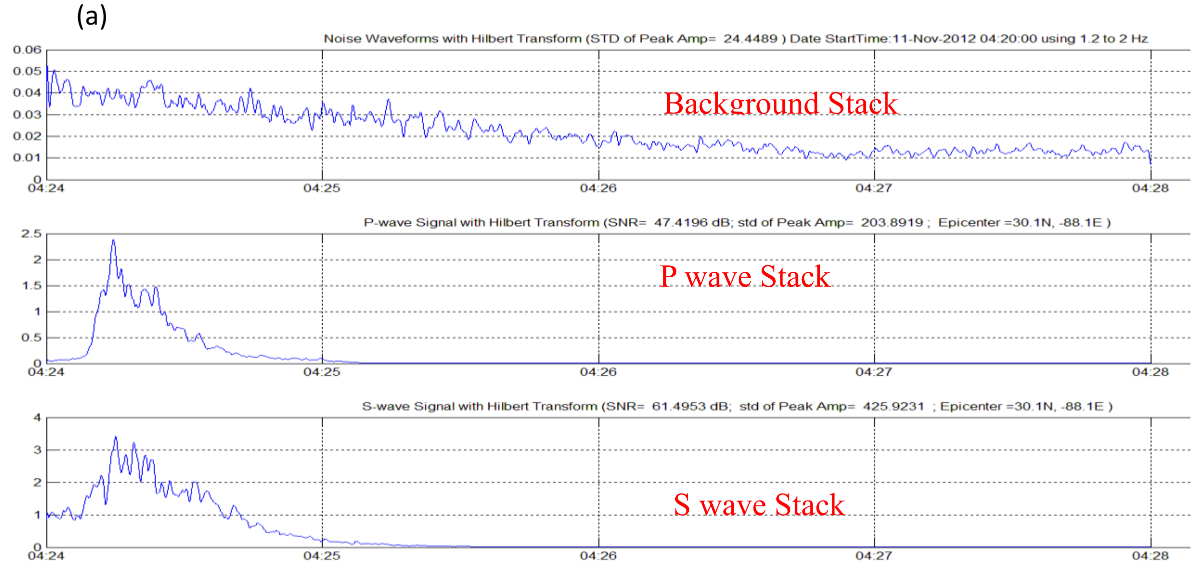


Figure 15: Effect of the RMS AGC correction on the TA waveforms of the Mississippi Nov 10, 2012 earthquake. (a) Filtered data before and (b) after the RMS AGC. It is shown that amplitudes at farther stations are boosted while amplitudes at nearer stations are reduced.



(a)



(b)

Figure 16: Weighted stacking of the background noise (top panel), P wave (middle panel) and S wave (bottom panel) using (a) the envelope functions and (b) without envelope function of the filtered waveforms. Destructive stacking is evident in the stacks of the P and S waves when the envelope function was not used. The P and S wave stacks when the envelope function was used show distinctive characteristics which are absent in the stack of the background hence an event is declared. The SNR in (a) without the Hilbert transform are -144.4 dB and 64.9 dB for P and S waves respectively. However, the SNR in (b) with the Hilbert transform are 47.4 dB and 61.5 dB for P and S waves respectively. The use of the Hilbert transform thus increases the SNR and hence enhances the detection of the small earthquakes.

$$S = \sum_{i=1}^{N} \frac{1}{std_j^2}\ H(a_{i,j}),\ \ j = (1,2,3, \dots, M) \ \dots\dots\dots\dots\dots\dots\dots\dots \text{(2.11)}$$

where std_j is the standard deviation of the background noise at the jth station. $H(a_{i,j})$ is the amplitude at ith position of the envelope of the seismic waveform from the jth TA station. S is the stacked seismic waveform within the data window.

An event is detected when the SNR and lambda of the P and S wave stacks are both greater than a chosen threshold value. The thresholds used in this research for the SNR and lambda are 10 and 2.5, respectively. SNR and lambda are given as:

$$SNR = 10\ \log_{10}(\frac{P_x}{P_v})\ dB \ \ \dots\dots\dots\dots\dots\dots\dots\dots \text{(2.12 a)}$$

$$Lambda = \frac{standard\ deviation\ of\ the\ signal}{standard\ deviation\ of\ the\ noise} \qquad \dots\dots \text{(2.12 b)}$$

where P_x and P_y represent the powers of the signal and background noise respectively. The power of a waveform is the sum of the squares of the amplitudes of the seismic waves in a data window.

When an event is declared by the event extraction algorithm, the 4-minute window of the seismic waveforms at all TA stations are displayed on a plot of the waveforms at each TA stations with respect to distance from the Gulf using an edited version of the code *section_display.m* of Mousa and Al-Shuhail (2009). The purpose of the plot is to visually classify the detected events into false detections, quarry blasts, local earthquakes, regional earthquakes or teleseismic using the observed P and S wave moveouts at the TA stations.

CHAPTER THREE: RESULTS

3.1 Introduction

Detected events can be classified into false detections, quarry blasts from the three Walter Minerals coal mines in Alabama, teleseismic events, regional earthquakes, earthquakes from nearby countries, and the local events which are the goal of this research (Fig. 17). In addition to the real events that were detected by the event extraction algorithm, false detections were also detected using the method of study.

Althought one of the goals of the detector is to minimize the number of false detections, such events were often detected by the event extraction algorithm due to the high-amplitude background noise at some TA stations that are located on undercompacted soils, especially those stations that are closest to the Gulf coast. However, unwanted detections can also be caused by manmade activities, e.g. detonations of dynamite at quarry sites that generate seismic waves and were detected by the TA stations as P and S waves. The waveforms from local earthquakes can be distinguished from regional earthquakes by a relatively smaller difference between the P and S wave arrival times at the TA stations nearest to the NGM.

The event extraction algorithm detected five new earthquakes in the NGM that were not in either the IRIS or USGS earthquake catalog (Fig 18). The dates of these new detected earthquakes were on August 20, 2012, August 26, 2012, September 3, 2012, October 4, 2012 and December 28, 2012.

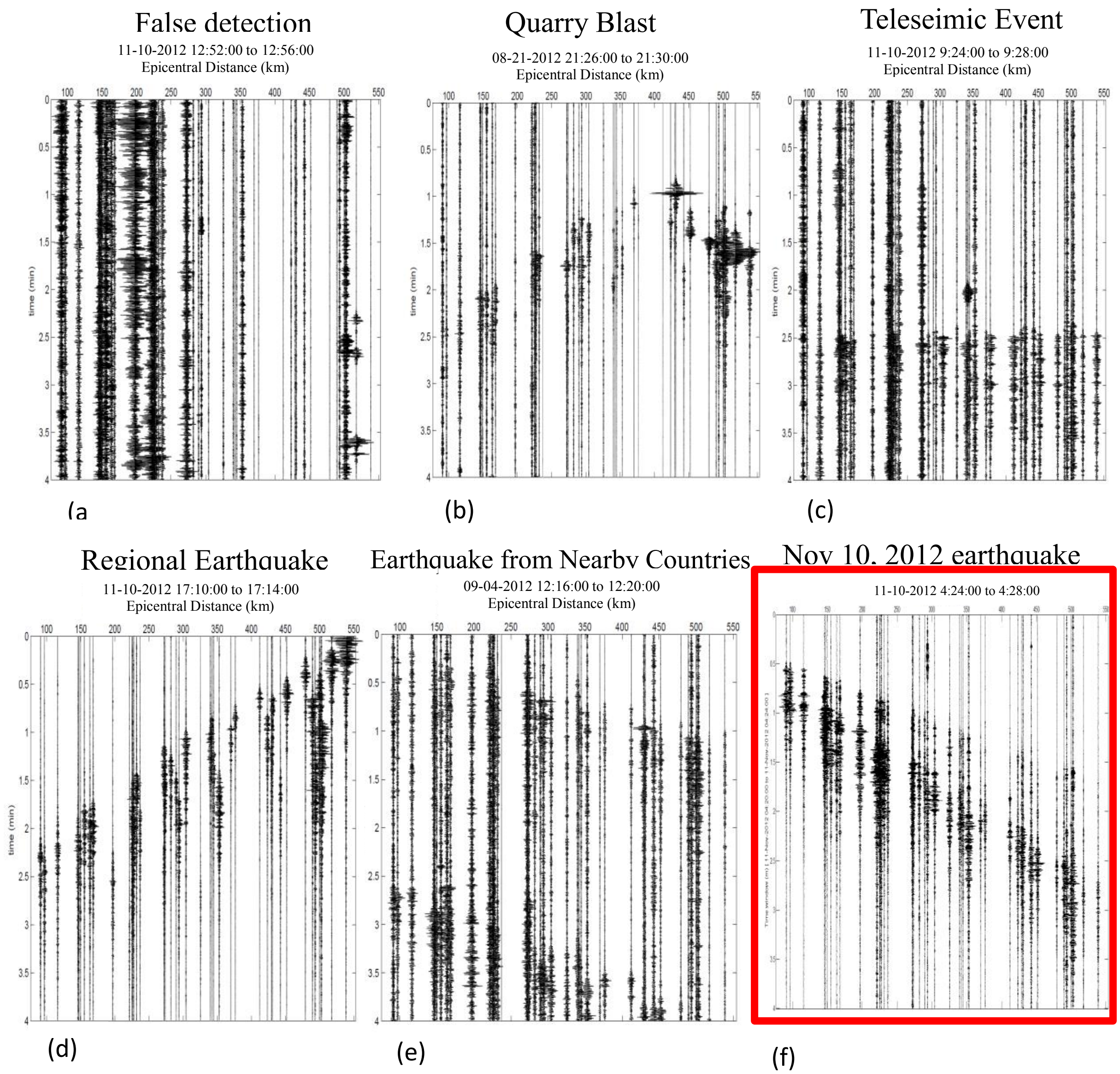


Figure 17: Plot of waveforms at different stations with distance away from Gulf showing the types of events detected between 2011 and 2013 in the NGM. (a) False event detection by the event extraction algorithm (b) Seismic section of quarry blast in the Walter Mineral Coal Mines. (c) A teleseism which shows the event reaching the stations in the NGM at nearly the same time. (d) An example of an event from Kentucky. (e) Earthquake from Mexico which shows both the P- and S- waves. (f) Seismic section of the Mississippi November 10, 2012 earthquake without the Automatic Gain Control (AGC). The red box indicates that the November 10 earthquake was known prior to this study.

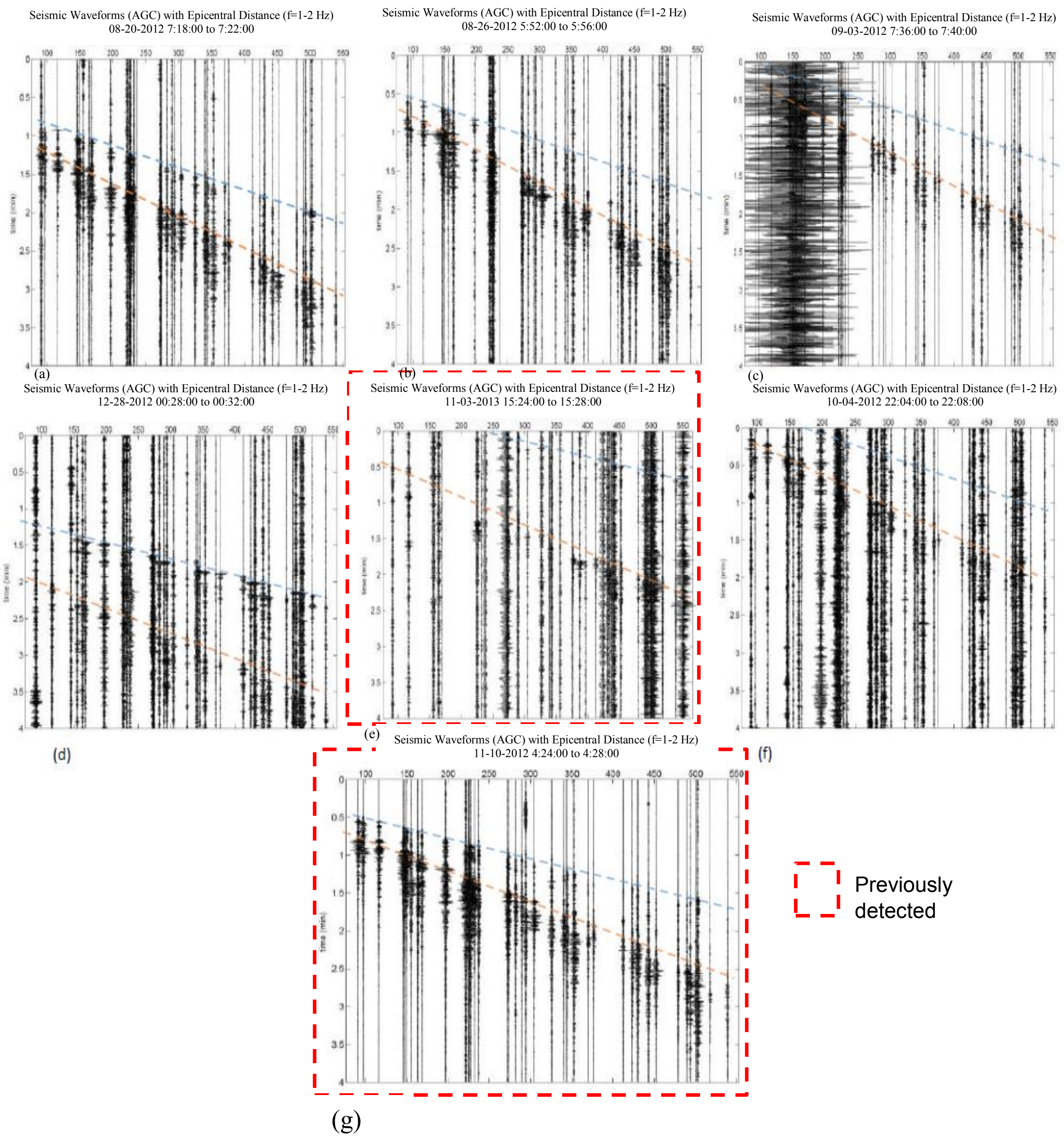


Figure 18: Newly detected earthquakes in the NGM showing the waveforms at all the TA stations with the same moveouts as the previously detected earthquakes in the NGM. (a) August 20, 2012 (b) August 26, 2012 (c) September 3, 2012. (d) December 28, 2012. (e) March 11, 2013. (f) October 4, 2012. (g) November 10 2012. The blue line shows the P-wave arrival across the stations while the red line shows the S-wave arrival.

3.2 Absolute Location, Origin Time and Depth of the Detected Earthquakes

The epicenters, origin times and focal depths of the newly detected earthquakes were determined using the HYPOINVERSE 2000 absolute location software (Klein, 2002). The inputs to the absolute location program are the P and S arrival times of the event at the TA stations. Six seismic velocity models of the NGM from the seismic refraction survey data across the USA (Fig. 19, Chulick and Mooney, 2002) were considered applicable for locating earthquakes in the NGM. These velocity models (Table 1) were used in the absolute epicenter determination to quantify the effect of the changes in velocity models on the absolute epicenter of the detected earthquakes.

After the absolute epicenters of the detected earthquakes were determined, a jackknife statistical technique was then applied to the arrival time data for each velocity profile to estimate the variance in the epicenter for the each earthquake. The jackknife statistical method involves determination of epicenters when the arrival time at one TA station is excluded. The removed arrival time then replaced and the process is repeated by removing the arrival time at another TA station. This method was used until all the TA stations have been excluded. This statistical technique creates a cloud of epicenters around the mean epicenter. The mean epicenter is the epicenter determined by the HYPOINVERSE 2000 when all the arrival times at all TA stations are used. Thereafter, the distance between each epicenter and the mean epicenter was determined, and the standard deviation of these distances was subsequently determined. Absolute locations of all of the epicentral determinations in the jackknife analysis were plotted to visualize the cluster of epicenters, and two concentric circles centered on the mean epicenter were plotted on the map. The radii of these circles are one and two standard deviations, which

Velocity Profile in the NGM

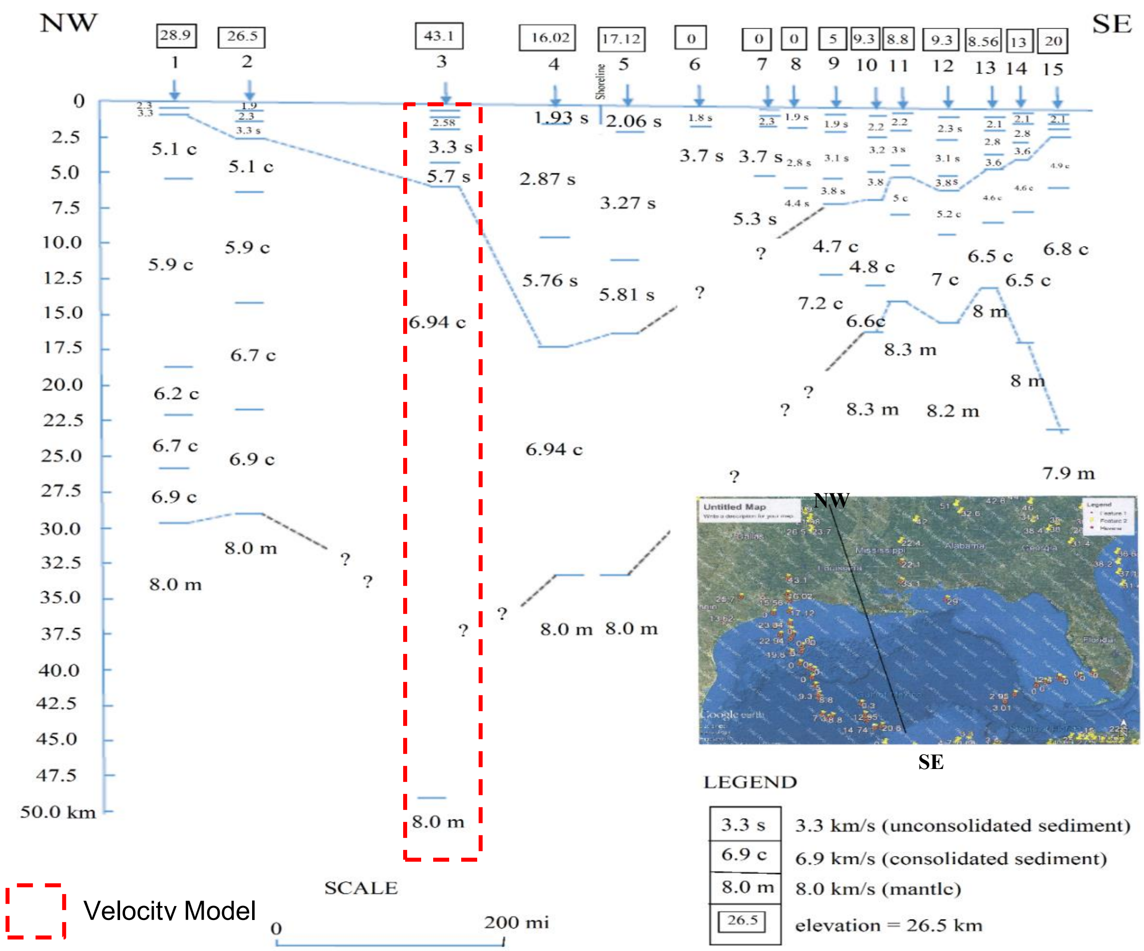


Figure 19: Velocity profile in the NGM with the line of the profile extending from onshore Louisiana to the deep offshore NGM (data extracted from Chulick and Mooney, 2002). The map at the bottom right shows a map view of the NGM with the line of the profile and the locations of seismic velocity profile from Chulick and Mooney (2002). The velocity profile shows three major layer groups: unconsolidated sediments (P wave velocity range of 1.79 – 5.84 km/s), consolidated sediments (P wave velocity range of 5.1 – 7.2 km/s and depth range of 1.0 – 17.5 km) and the upper mantle rocks (P wave velocity range of 7.9 – 8.3 km/s and depth range of 15.0 – 33.0 km). The red dash box shows an example of how model 2, which was used in the determination of absolute location of the earthquakes, was extracted from the velocity profile of the NGM.

Table 1: Seismic velocity models used for the determination of absolute locations of the detected earthquakes in the NGM (data extracted from Chulick and Mooney, 2002)

Rock Types	P wave Velocity (km/s)	Depth (km)	P wave Velocity (km/s)	Depth (km)
	Model 1		Model 2	
Unconsolidated sediments	2.3	0.0	2.5	0.0
			3.3	1.0
Consolidated sediments	5.1	1.0	5.7	4.0
	5.9	5.5	6.9	6.0
	6.2	18.5		
	6.7	22.0		
	6.9	26.0		
Upper mantle	8.0	29.0	8.0	32
	Model 3		Model 4	
Unconsolidated sediments	1.9	0.0	2.1	0.0
	2.9	1.5	3.3	2.0
Consolidated sediments	5.8	10.0	5.8	10.5
	6.9	17.5	6.7	16.0
Upper mantle	8.0	32.5	8.0	33.0
	Model 5		Model 6	
Unconsolidated sediments	2.1	0.0	1.7	0.0
	2.3	1.0	2.1	0.5
	3.1	2.5	2.8	1.0
	3.8	5.0	3.6	3.5
			4.6	6.0
Consolidated sediments	5.2	6.0	6.5	9.5
	7.0	10.0		
Upper mantle	8.0	15.0	8.0	17.5

represent the 68% and 95% confidence intervals in the epicentral determination, respectively, based on the jackknife analysis. Table 2 shows the radii of the 68% certainty in epicenter for the six velocities models for the detected earthquakes. The locations computed using velocity model 1 have the lowest uncertainty in the epicenter locations for all the detected earthquakes except for the December 28, 2012 and March 11, 2013 earthquakes and was used for the absolute epicenter. Model 2 was used for the December 28, 2012 and March 11, 2013 earthquakes because it has the lowest uncertainty in the epicenter of these earthquakes. The results of the HYPOINVERSE 2000 are presented in Table 3.

The August 20th, 2012, August 26, 2012, September 3, 2012 and November 10, 2012 earthquakes are both located in the offshore part of the Mobile Bay (Fig. 20, 21, 22 and 23, respectively, Appendix C). Conversely, The October 4, 2012 earthquake is located onshore near Baton Rouge in Louisiana; the March 11, 2013 earthquake occurred near New Orleans (a previously known event) while the December 28, 2012 earthquake occurred far offshore and their epicenters were poorly constrained because of ground reverberations due to the looseness of the sediments (Fig. 24, 25 and 26, respectively). The focal depths of these earthquakes were poorly determined because there is no seismic station close to the epicenter. One or more stations are required by HYPOINVERSE 2000 software to be above the hypocenter in order to constrain the depth of the event.

Table 2: 68% certainty in earthquake determinations using different velocity models.

Earthquakes	Model 1 (km)	Model 2 (km)	Model 3 (km)	Model 4 (km)	Model 5 (km)	Model 6 (km)
August 20, 2012	2.06	2.49	9.2	4.05	3.39	6.04
August 26, 2012	1.5	2.11	1.68	1.59	91.7	4.92
September 3, 2012	1.318	3.65	1.408	2.837	7.82	11.33
October 4, 2012	90.8	107.9	133.9	135.8	120.8	92.79
November 10, 2012	5.33	5.0	2.67	3.56	7.02	7.95
December 28, 2012	38.16	12.5	15.3	14.86	13.16	13.35
March 11, 2013	450	54.8	410.2	17.4	381.9	440.1

Table 3: Classification of detected earthquakes and their locations, depths, magnitudes and the uncertainties. The magnitudes, in parenthesis, represent the magnitudes determined by USGS.

SN	Classification of events	Earthquake	Origin time	Lat (N)	Lon (W)	Depth (km)	M lg	Standard deviation of M lg
1	New	August 20, 2012	7:18:36.88	30.16^0	87.99^0	0.97	2.00	0.3
2	New	August 26, 2012	5:52:18.00	30.24^0	87.96^0	2.11	1.99	0.34
3	New	September 3, 2012	7:35:48.95	30.31^0	87.95^0	6.6	2.19	0.21
4	New	October 4, 2012	22:03:14.09	30.47^0	90.80^0	4.83	2.21	0.22
5	Previously detected	November 10, 2012	4:24:14.28	30.19^0	88.03^0	0.89	2.58 (2.6)	0.13
6	New	December 28, 2012	0:28:28.23	27.60^0	88.41^0	5.0	2.77	0.22
7	Previously detected	March 11, 2013	15:23: 9.14	28.84^0	90.48^0	10.0	3.03 (2.9)	0.28

Epicenter Determination of the August 20, 2012 Earthquake

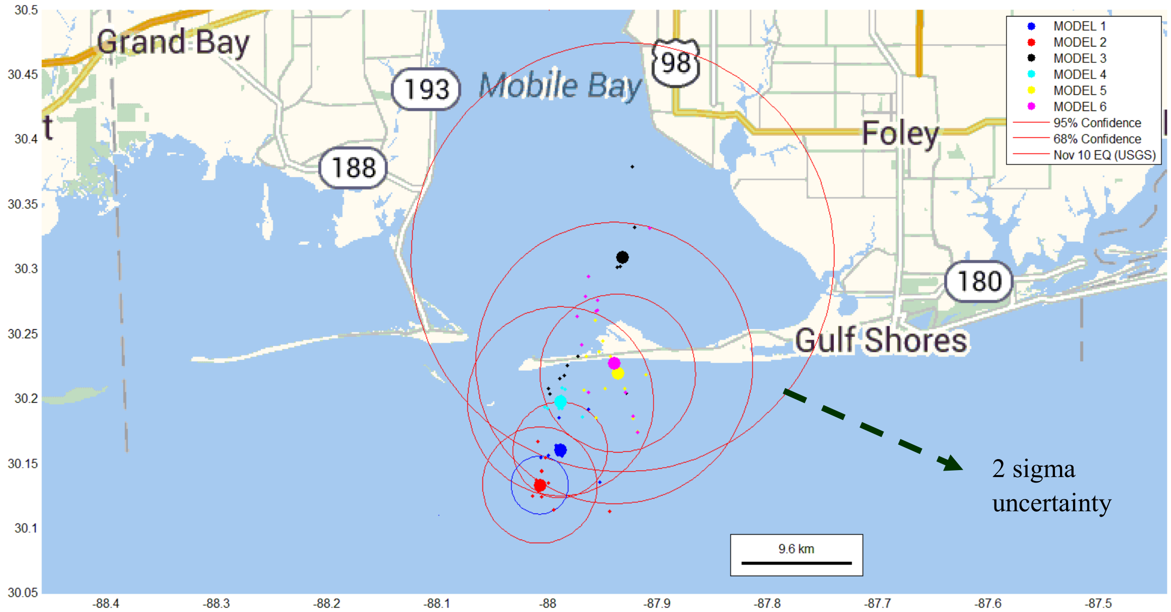


Figure 20: Epicenters and their uncertainties of the August 20, 2012 earthquake using six different velocity models for the absolute locations. Epicenters are the colored points with the velocity models used differentiated by the colors. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 1), is the best estimate. Other circles of uncertainties are omitted in subsequent plots.

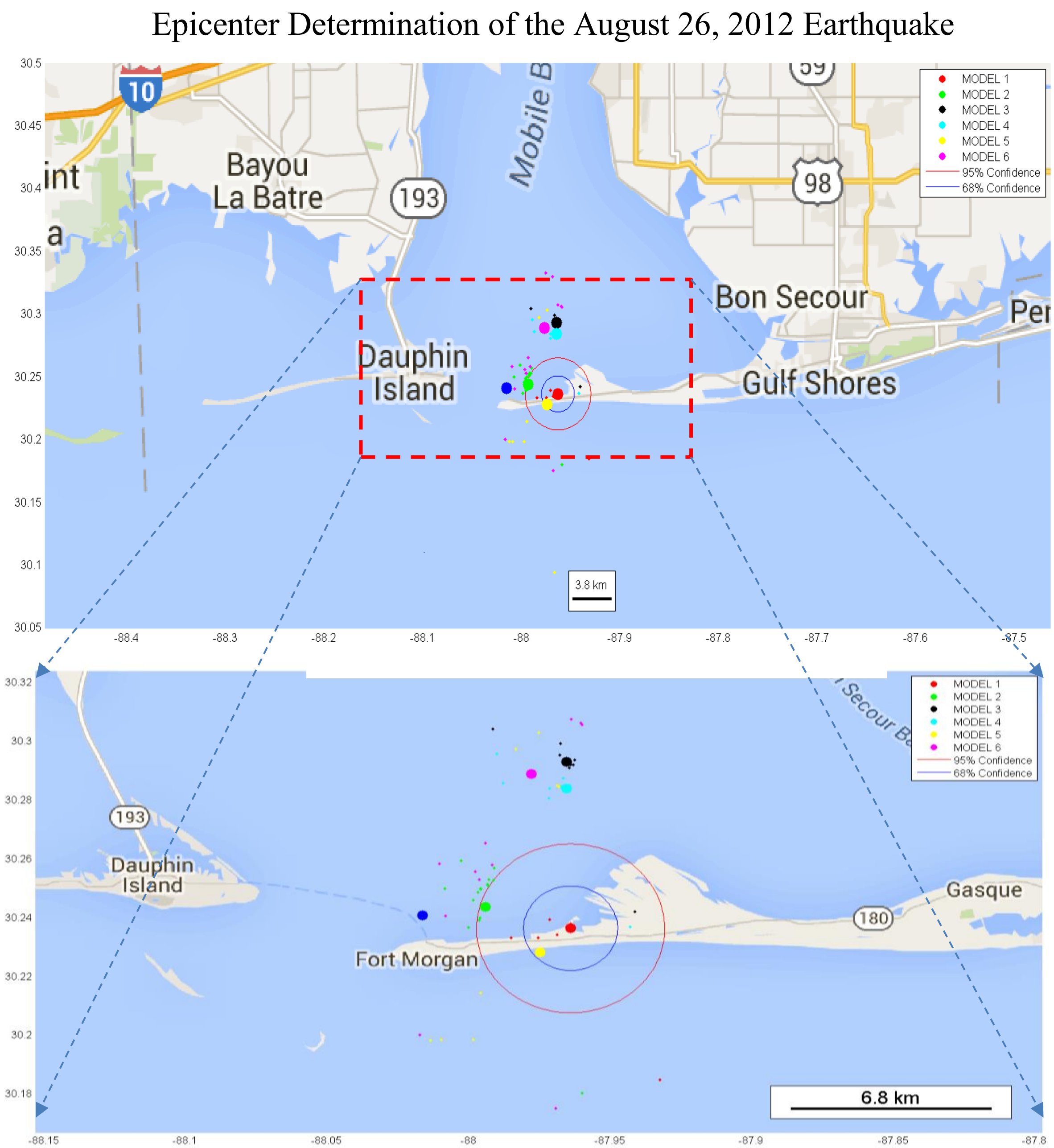


Figure 21: Epicenters and their uncertainties of the August 26, 2012 earthquakes using six different velocity models. Epicenters are mostly concentrated within the Mobile Bay area and are represented by the colored points with each color representing the velocity model used. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 1), is the best estimate.

Epicenter Determination of the September 3, 2012 Earthquake

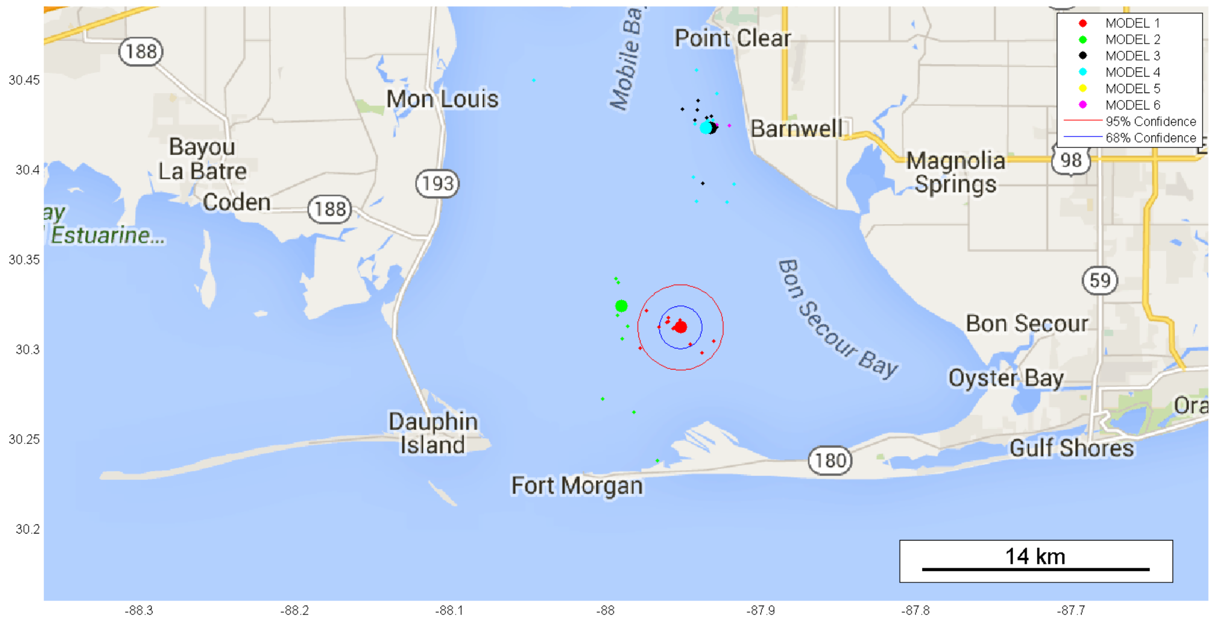


Figure 22: Epicenters and their uncertainties of the September 3, 2012 earthquakes using six different velocity models. Epicenters are the colored points and each color represents the velocity model used. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination.

Epicenter Determination of the October 4, 2012 Earthquake

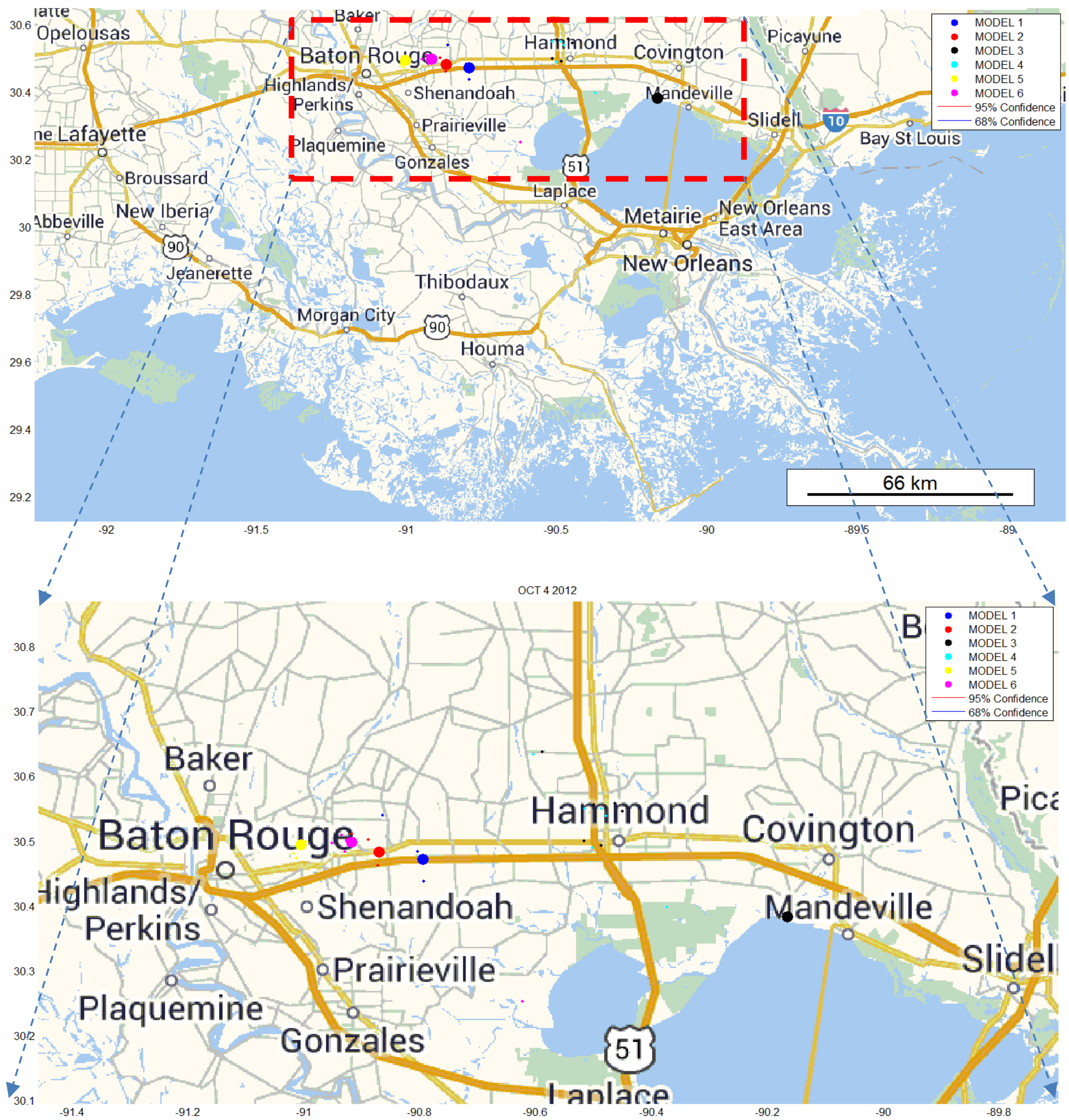


Figure 23: Epicenters and their uncertainties of the October 4, 2012 earthquakes using six different velocity models. Epicenters are the colored points and each color represents the velocity model used and the epicenters are concentrated onshore around Baton Rouge. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 1), is the best estimate.

Epicenter Determination of the November 10, 2012 Earthquake

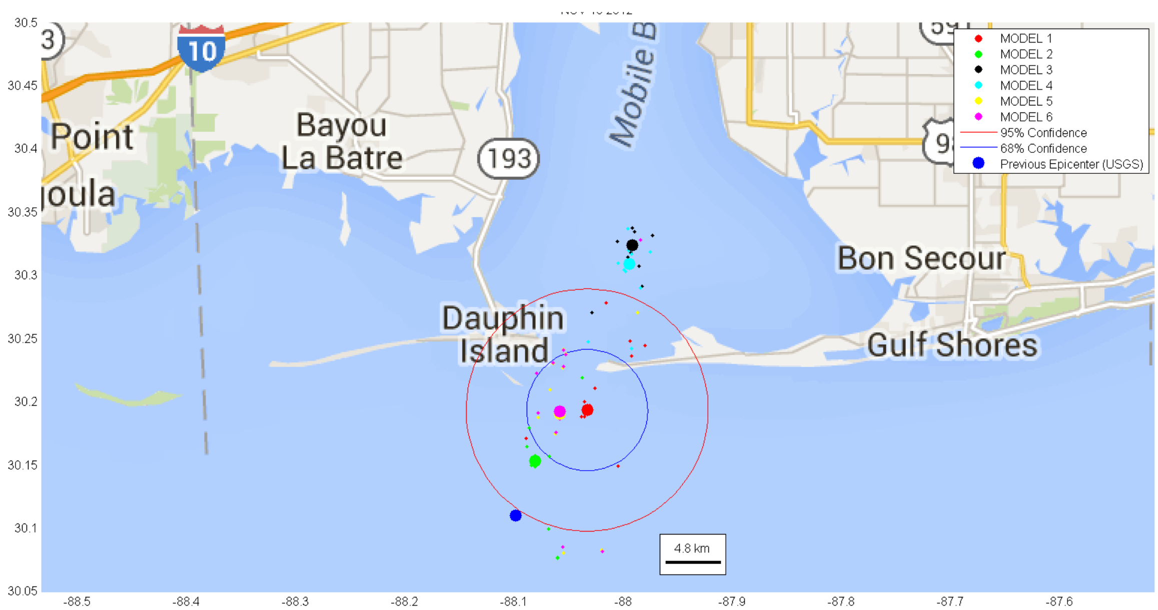


Figure 24: Epicenters and their uncertainties of the November 10, 2013 earthquakes using six different velocity models. Epicenters are the colored points and each color represents the velocity model used. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 1), is the best estimate.

Epicenter Determination of the December 28, 2012 Earthquake

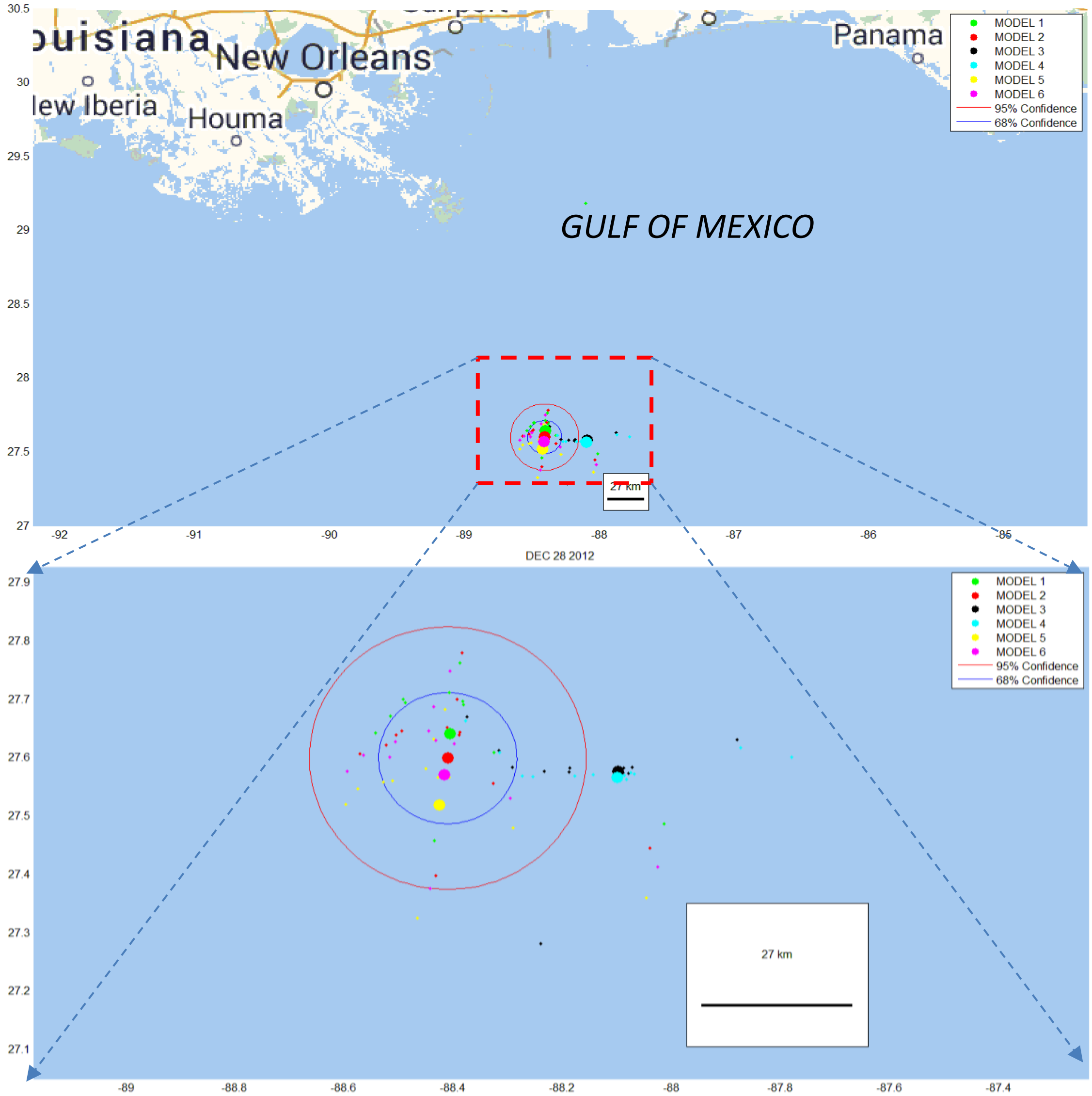


Figure 25: Epicenters and their uncertainties of the December 28, 2012 earthquakes using six different velocity models. Epicenters are the colored points and each color represents the velocity model used. The epicenters are concentrated in the far offshore NGM so may not be related to the gas extraction in the Mobile Bay area. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 2), is the best estimate.

Epicenter Determination of the March 11, 2013 Earthquake

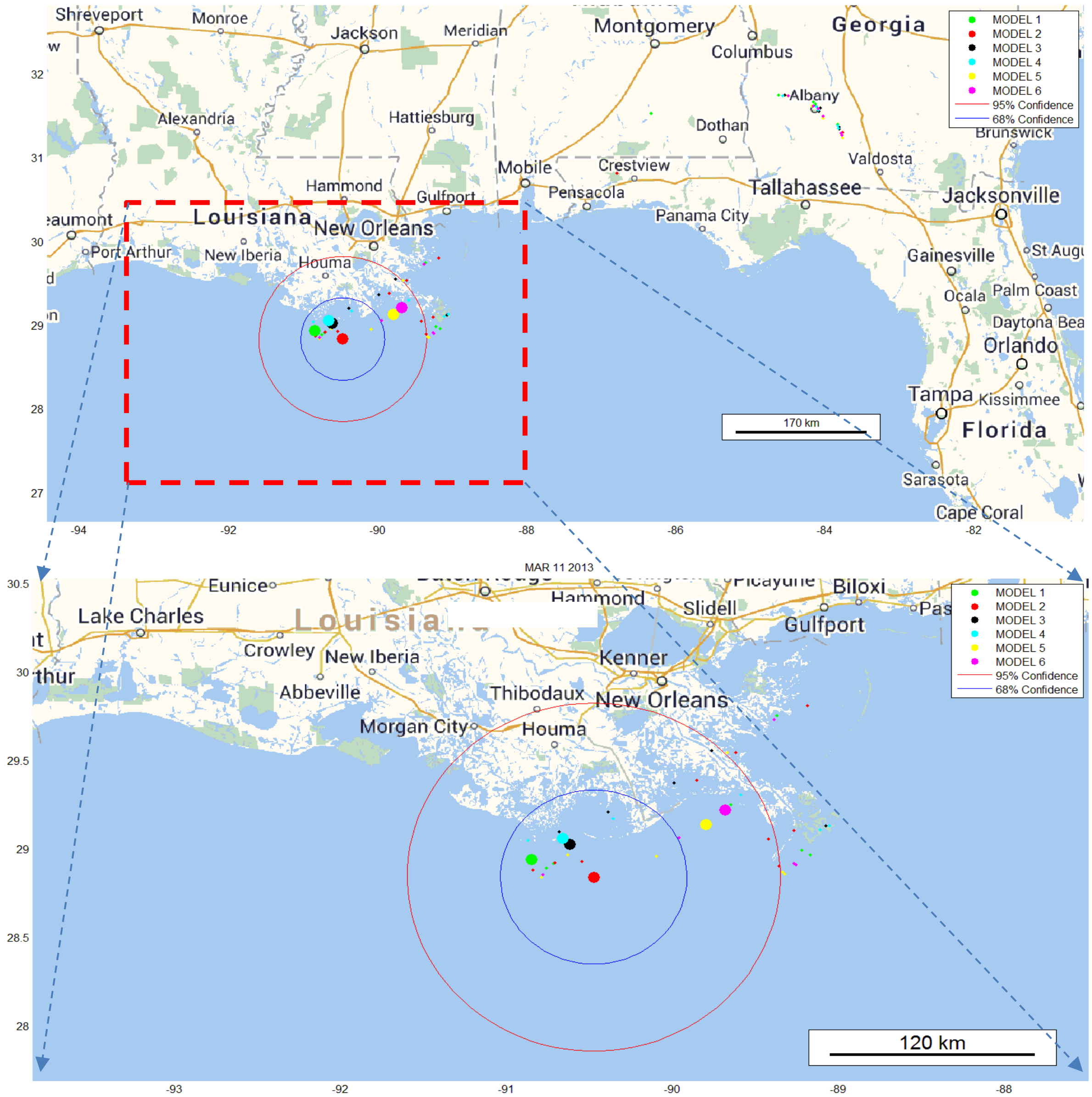


Figure 26: Epicenters and their uncertainties of the March 11, 2013 earthquakes using six different velocity models. Epicenters are the colored points and each color represents the velocity model used. The mean epicenters are represented by bigger points and show the computed locations when the arrival times at all TA stations were used. The smaller points represent various epicenters from the jackknife statistical technique, i.e., the epicenters when each arrival times was removed. The blue and red circles show 68% and 95% confidence intervals in the epicentral determination. The epicenter with smallest radius, and hence high confidence level (i.e., Model 2), is the best estimate.

3.3 Magnitudes (Mlg) determination

The magnitude of the detected earthquakes is an important property that is necessary to estimate possible hazard. Earthquake magnitude were determined in this study using Herrmann and Nuttli's equation (Ebel, 1994).

$$Mlg(f) = X + \text{nlo g}\left(\frac{r}{10}\right) + \ logA + 0.4342\ gamma\ (f) * r \quad \text{……. (2.13 a)}$$

$$gamma\ (f) = \ \frac{\pi f}{Q\ (f) * 3.55\ km/s} \quad \text{…………………… (2.13 b)}$$

$$Q = 365 * f^{0.624} \quad \text{(Chapman and Pezeshk, 2014)…. (2.13 c)}$$

The values of X and n in the magnitude determination formula are different from one geologic setting to another. These values were determined for the NGM by plotting the right hand side terms of equation 2.15 with respect to log(r/10) at all TA stations (Fig. 27) using the frequency, amplitudes, epicentral distance of the TA stataions and magnitude of the known November 10, 2012 earthquake. This plot is a linear graph with intercept X and slope n. The straight line graph has a correlation coefficient of 0.7996, and the values of X and n were estimated to be -1.6246 and 0.6212, respectively.

$$Mlg(f)\ - \left(logA + 0.4342\ gamma\ (f) * r\right) = X + \text{nlog}(\frac{r}{10}) \quad \text{……….... (2.15)}$$

The magnitude of each earthquake was be determined using the mean of the magnitudes determined using the seismic waveforms from the TA stations. The standard deviation of the estimated magnitudes from each TA station is the uncertainty in magnitude . The value of the standard deviation for the estimated magnitudes for the five

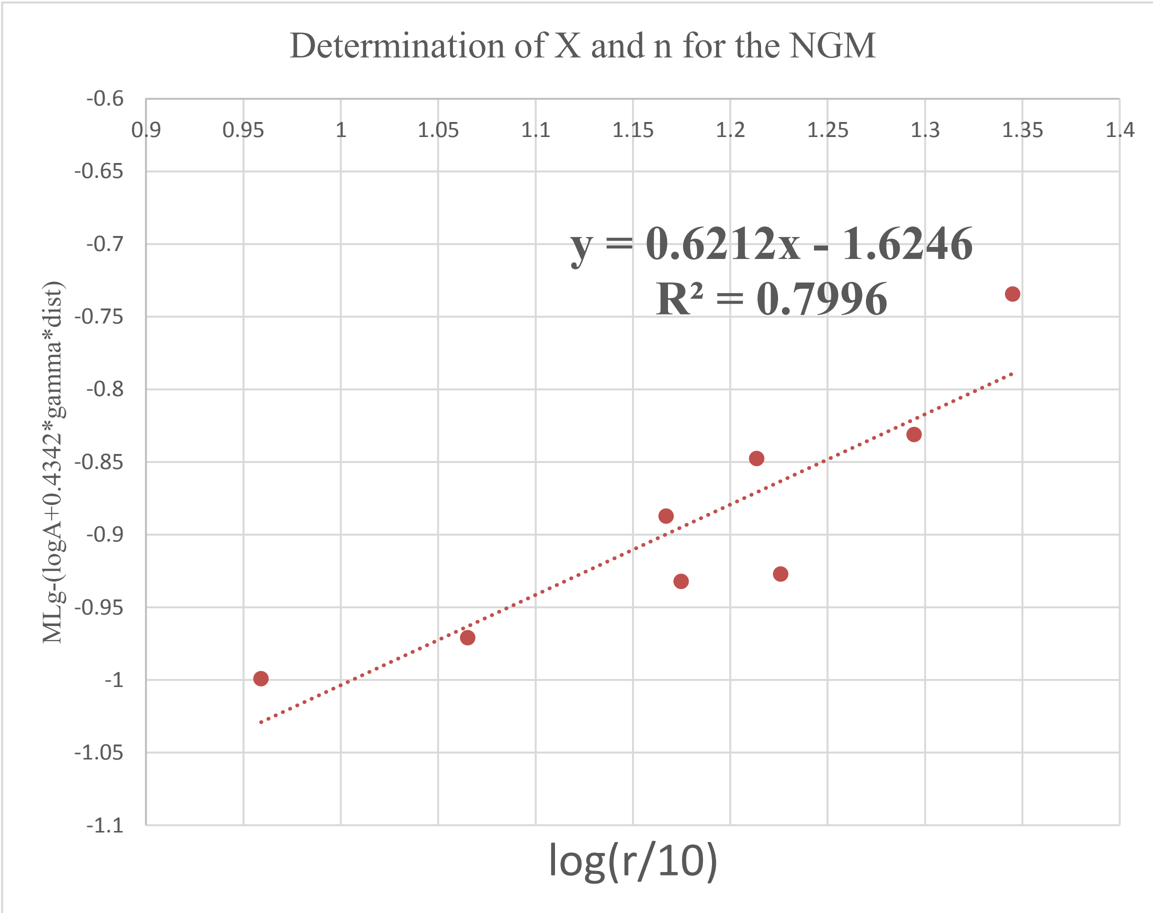


Figure 27: Determination of X and n for the NGM to estimate the magnitude of the detected earthquakes. The figure shows a plot of the right hand terms of the equation $Mlg(f) - \left(logA + 0.4342\, gamma\, (f) * r\right) = X + \mathrm{nlog}(\frac{r}{10})$ against log(r/10). The plot is expected to be a straight line with slope as n and intercept X. Mlg = Lg magnitude, r = epicentral distance, A = peak amplitude (Amax – Amin)/2, f = frequency and X is the constant that calibrates the absolute level of the magnitude scale; n is the exponent of the geometric spreading function for Lg waves using 3.55 km/s is the group velocity. X and n are site specific constants. Gamma is frequency dependent and also depends on crustal anelastic attenuation coefficient, Q, which is site dependent.

newly detected earthquakes and two previously detected earthquakes ranges from 0.13 to 0.34. The sources of error in the magnitude determinations include errors from peak amplitude, frequency and epicentral distance determinations, the geology of the site which affects the amplitudes of the waveforms due to the P and S wave velocities of the rocks at the TA stations (Fig. 28). The cumulative effect of these sources causes a bias at the TA stations in the magnitude (Mlg) determination (Fig. 29).

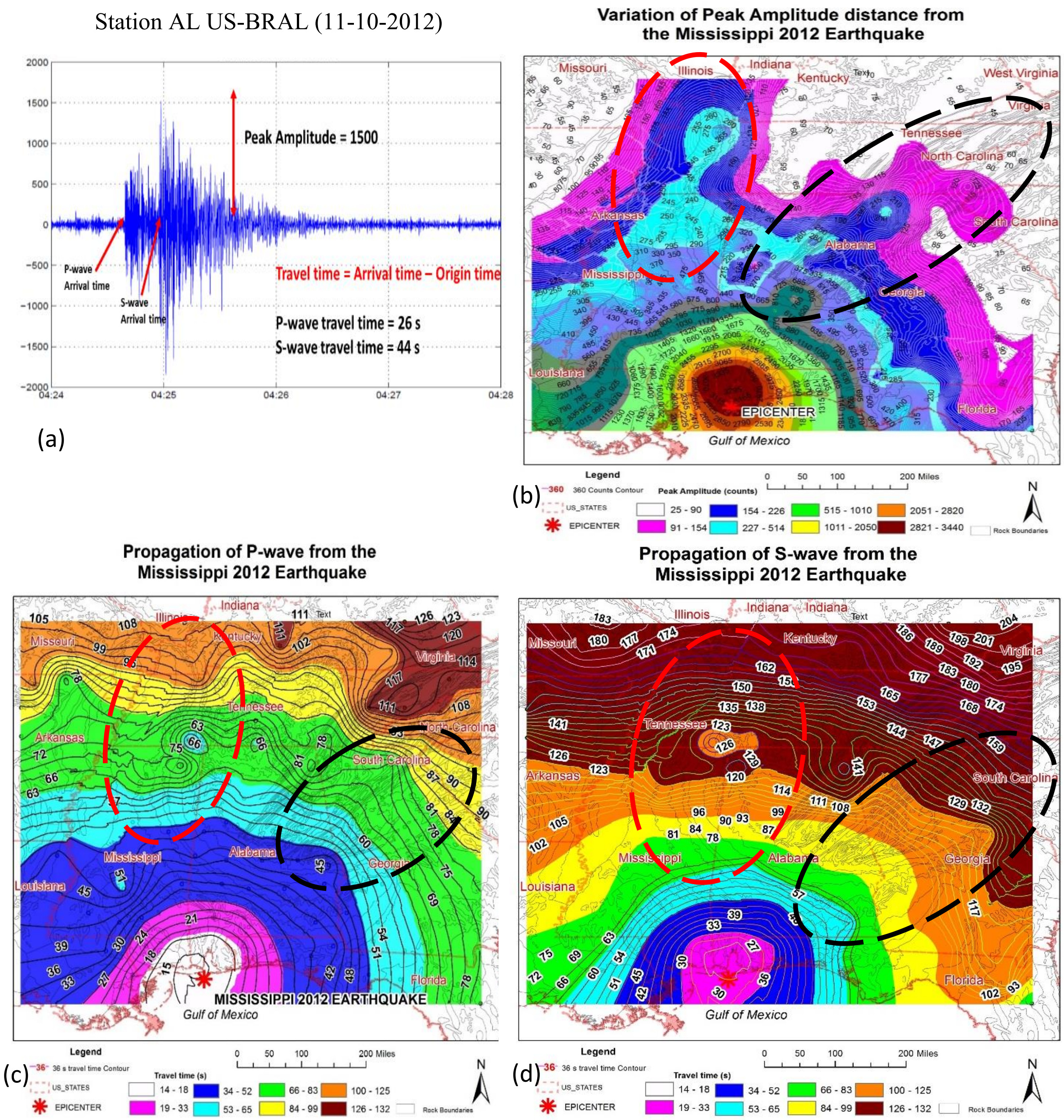
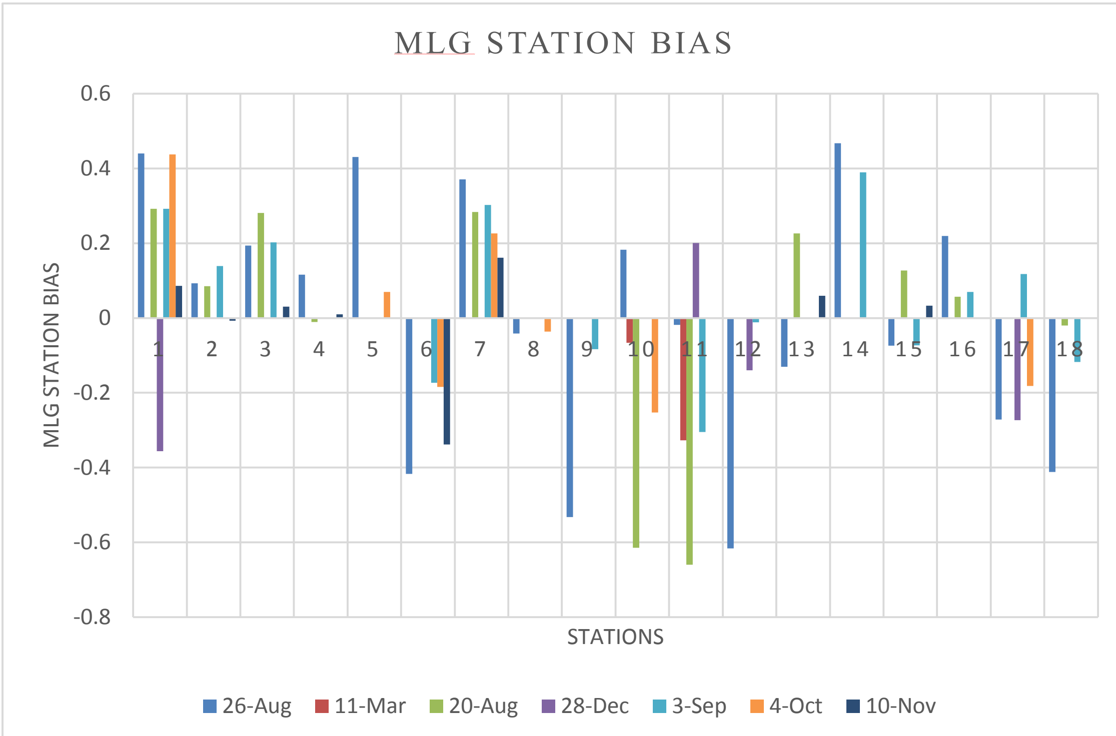


Figure 28: Effect of geology on propagation of P and S wave arrival times and the peak amplitudes which in turn affect the determination of magnitudes (a) Seismic waveform from the November 10, 2012 earthquake detected at AL US-BRAL station in Alabama. (b) Variation of peak amplitude due to dissipation of energy as seismic wave propagates from the epicenter. (c - d) Propagation of P- and S- waves from the Mississippi 2012 earthquake. The propagation pattern is changed by the Mississippi Embayment (surrounded by red dashed circle) and the southern Appalachians (surrounded by the black dashed circle). 100 of the 502 TA stations operating at the time of the earthquake were used because amplitudes at the other 402 TA stations, farther from the epicenter, have decreased below the background noise at seismic stations.



1	349A	7	248A	13	347A
2	448A	8	250A	14	348A
3	449A	9	Y48A	15	346A
4	446A	10	Z50A	16	450A
5	149A	11	Z49A	17	Y49A
6	249A	12	X47A	18	351A

Figure 29: Effect of station bias on magnitude determination. The station bias was calculated by subtracting the magnitude determined using one station from the mean magnitude determined using 18 TA stations with high SNR. Stations 349A, 248A, 348A have positive biases; stations 249A, Z49A and Y49A have negative biases while the rest have a combination of both.

CHAPTER FOUR: DISCUSSION

4.1 Spatial Correlation of the Detected Earthquakes and Oil and Gas Extraction

The spatial distributions of earthquakes were correlated with wells drilled in the Mobile Bay region, NGM, to determine if the detected earthquakes correlated with gas extraction from local wells (Fig. 30 and 31, Table 4). All drilled well locations and status information are from the Bureau of Safety and Environmental Enforcement (BSEE) (2014). Wells in Mobile Bay area are primarily for gas exploitation. As such their statuses are classified as either dry and abandoned, plugged and abandoned, producing, shut in and temporarily abandoned wells. The focus of the correlation was on Mobile Bay because the epicenters of the two previously known earthquakes and epicenters of three of the new earthquakes are located in the area.

Histograms of the number of wells around each detected earthquakes show that all of the detected earthquakes (i.e., August 20, 2012, August 26, 2012, September 3, 2012 and November 10, 2012) in the Mobile Bay area occurred within 10 km of producing wells (Fig. 32, 33, 34 and 35, repectively). For model 1, the percentage of the producing wells within 10 km of the epicenters of the August 20, 2012, August 26, 2012, September 3, 2012 and November 10, 2012 earthquakes are 45.5%, 59.1%, 31.8% and 68.2%, respectively (Table 5). Most of the plugged and abondaned wells, dry and abandoned wells, shut in and temporarily abandoned are evenly distributed within 35 km of the epicenters while the locations of the producing wells are anomalous, and hence these wells likely did not induce the detected earthquakes.

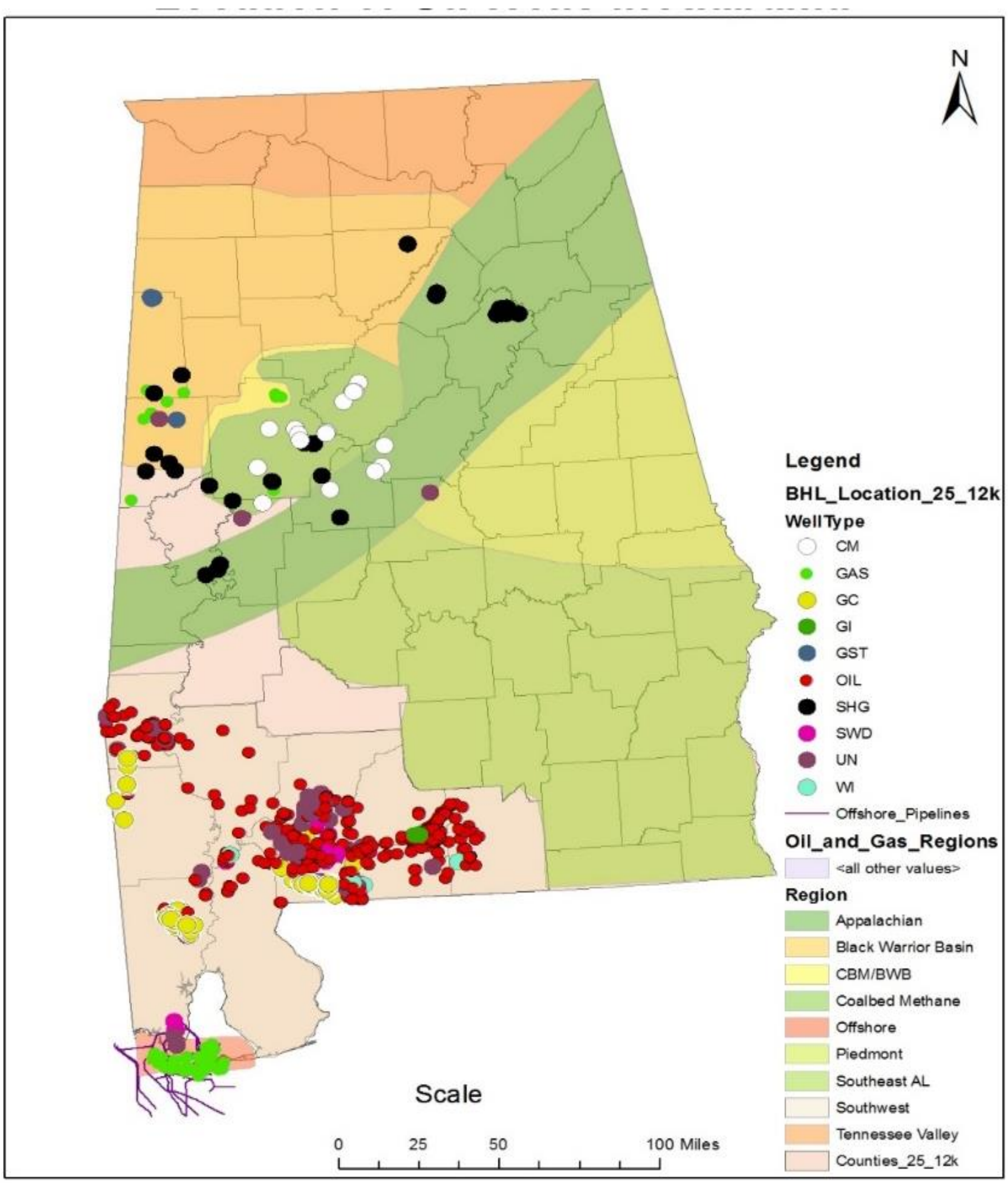


Figure 30: Location of oil and gas wells in Alabama showing the different types of production with different symbols. The various regions are also colored differently. The gas producing wells are concentrated in the lower part of the map in the offshore region of the Mobile Bay (Data from BSEE, 2014).

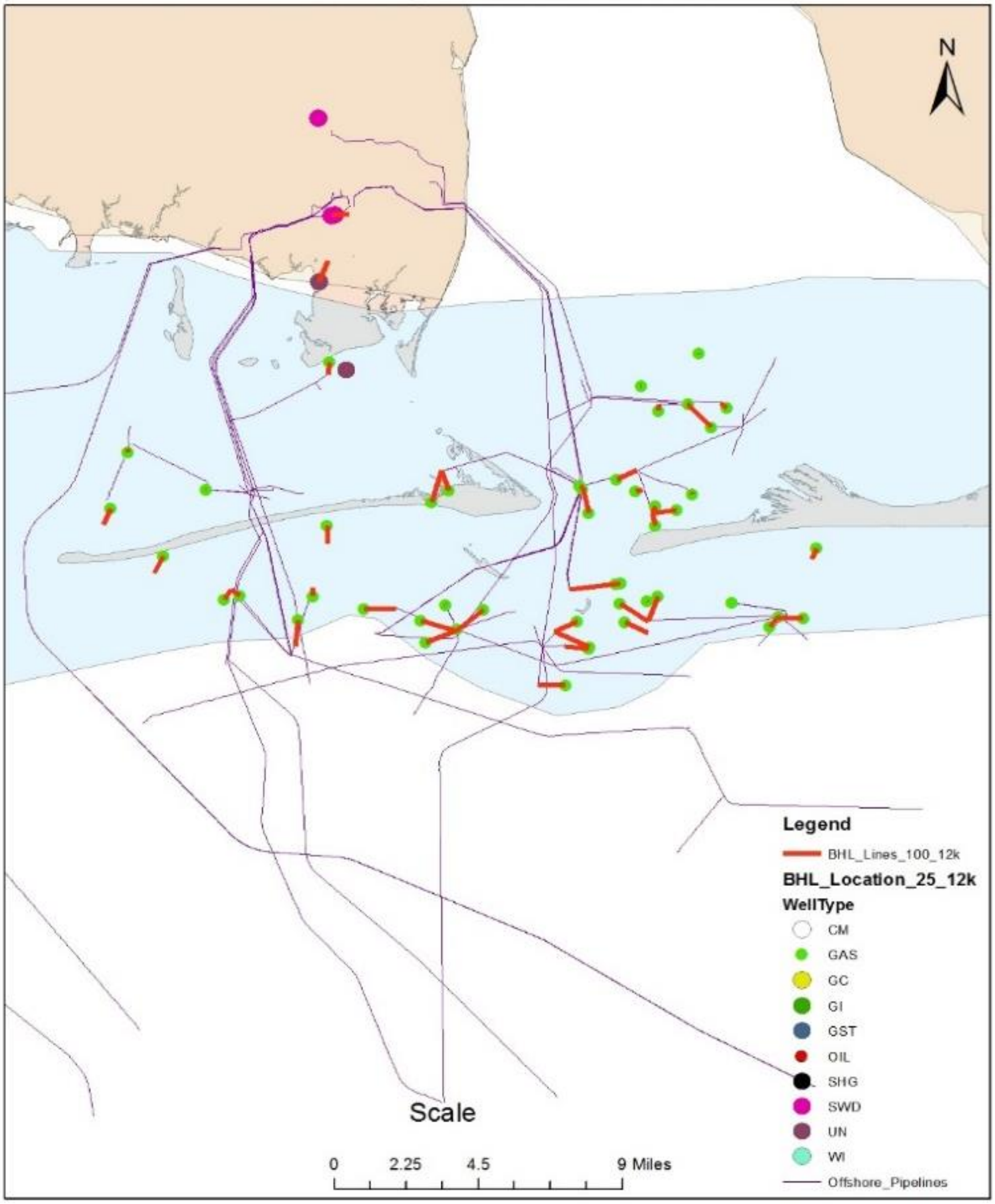


Figure 31: Locations of gas wells in the Mobile Bay. Red lines show the directions of directional wells while the purple lines represent the various pipelines in this area (data from BSEE, 2014)

Table 4: Drilled wells around the Mobile Bay area. All of the wells are for gas exploitation (BSEE, 2015)

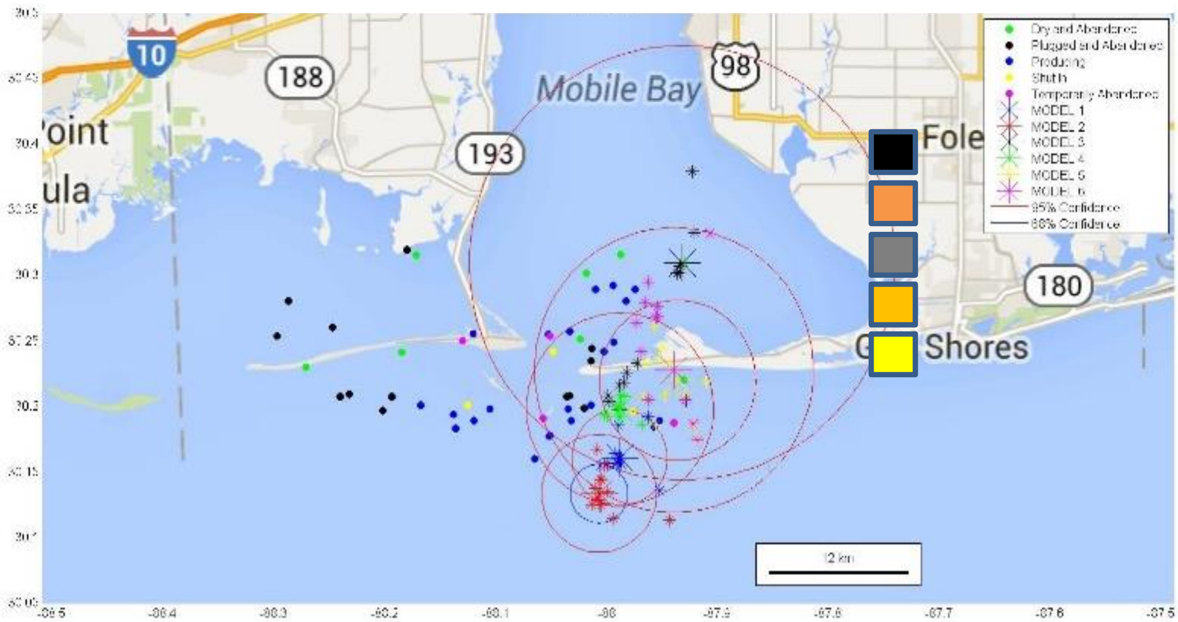
FID	Permit	Longitude	Latitude	API Number	Well Type	Classification
1	4576-OS-27	-87.9885	30.31562	1.2032E+12	GAS	Dry and Abandoned
2	11156-OS-66-B	-88.173	30.31494	1.2972E+12	UN	Dry and Abandoned
3	4815-OS-30-B	-88.2723	30.22955	1.2972E+12	GAS	Dry and Abandoned
4	12103-OS-84	-88.0194	30.30131	1.2032E+12	GAS	Dry and Abandoned
5	5180-OS-31-B	-88.0247	30.25105	1.2032E+12	GAS	Dry and Abandoned
6	4600-OS-28-B	-87.9321	30.22028	1.2032E+12	GAS	Dry and Abandoned
7	4553-OS-26-B	-88.1861	30.24068	1.2972E+12	GAS	Dry and Abandoned
8	10008-OS-52-BH	-88.2987	30.25311	1.2972E+12	GAS	Plugged and Abandoned
9	10557-OS-59-B	-88.0374	30.20731	1.2972E+12	GAS	Plugged and Abandoned
10	4436-OS-24	-88.0213	30.19849	1.2032E+12	GAS	Plugged and Abandoned
11	5315-OS-33-B	-88.2035	30.19652	1.2972E+12	GAS	Plugged and Abandoned
12	10737-OS-61-B	-88.181	30.31899	1.2972E+12	GAS	Plugged and Abandoned
13	9597-OS-43-B	-88.0153	30.23444	1.2032E+12	GAS	Plugged and Abandoned
14	10557-OS-59-B1	-88.0348	30.20767	1.2972E+12	GAS	Plugged and Abandoned
15	9980-OS-45-B	-88.2415	30.20726	1.2972E+12	GAS	Plugged and Abandoned
16	10394-OS-55-B	-88.1952	30.20728	1.2972E+12	GAS	Plugged and Abandoned
17	3346-OS-8-B	-88.015	30.24364	1.2032E+12	GAS	Plugged and Abandoned
18	8384-OS-40-B	-88.2484	30.2601	1.2972E+12	GAS	Plugged and Abandoned
19	10954-OS-64-B	-88.2883	30.27976	1.2972E+12	GAS	Plugged and Abandoned
20	10414-OS-56-B	-88.2332	30.20901	1.2972E+12	GAS	Plugged and Abandoned
21	11009-OS-65-B	-88.1397	30.1936	1.2972E+12	GAS	Producing
22	14694-OS-95-B	-87.9951	30.24848	1.2032E+12	GAS	Producing
23	11180-OS-67-B	-88.0357	30.19788	1.2032E+12	GAS	Producing

24	11434-OS-78-B	-88.1068	30.19756	1.2972E+12	GAS	Producing
25	3840-OS-19-B	-88.0344	30.25702	1.2972E+12	GAS	Producing
26	6109-OS-34-B	-87.984	30.27999	1.2032E+12	GAS	Producing
27	12005-OS-83-B	-88.0157	30.2006	1.2032E+12	GAS	Producing
28	7080-OS-35-B	-88.0038	30.24153	1.2032E+12	GAS	Producing
29	4131-OS-22	-88.121	30.18878	1.2972E+12	GAS	Producing
30	12309-OS-86-B	-87.9753	30.28901	1.2032E+12	GAS	Producing
31	9962-OS-44-B	-88.122	30.25533	1.2972E+12	GAS	Producing
32	11825-OS-82-B	-88.011	30.28909	1.2032E+12	GAS	Producing
33	4477-OS-25-B	-88.053	30.17705	1.2972E+12	GAS	Producing
34	12393-OS-88-B	-87.9582	30.18386	1.2032E+12	GAS	Producing
35	9863-OS-46-B	-88.0335	30.18876	1.2032E+12	GAS	Producing
36	11293-OS-68-B	-88.1376	30.18315	1.2972E+12	GAS	Producing
37	12340-OS-87-B	-88.0524	30.17772	1.2972E+12	GAS	Producing
38	10211-OS-54-B	-88.1692	30.20045	1.2972E+12	GAS	Producing
39	4266-OS-23	-87.9535	30.18862	1.2032E+12	GAS	Producing
40	13485-OS-94-B	-88.0537	30.25473	1.2972E+12	GAS	Producing
41	5210-OS-32-B	-88.066	30.15977	1.2972E+12	GAS	Producing
42	4068-OS-21	-87.9953	30.29195	1.2032E+12	GAS	Producing
43	3135-OS-6-B	-88.0499	30.24157	1.2972E+12	GAS	Shut In
44	12155-OS-85	-88.1263	30.20077	1.2972E+12	GAS	Shut In
45	15847-OS-96	-87.9775	30.19607	1.2032E+12	GAS	Shut In
46	9477-OS-41-B	-88.0585	30.19037	1.2972E+12	GAS	Temporarily Abandoned
47	9985-OS-51-B	-87.9404	30.18717	1.2032E+12	GAS	Temporarily Abandoned
48	2543-OS-3-B	-88.0525	30.25342	1.2972E+12	GAS	Temporarily Abandoned
49	12604-OS-90-B	-88.1312	30.25004	1.2972E+12	GAS	Temporarily Abandoned

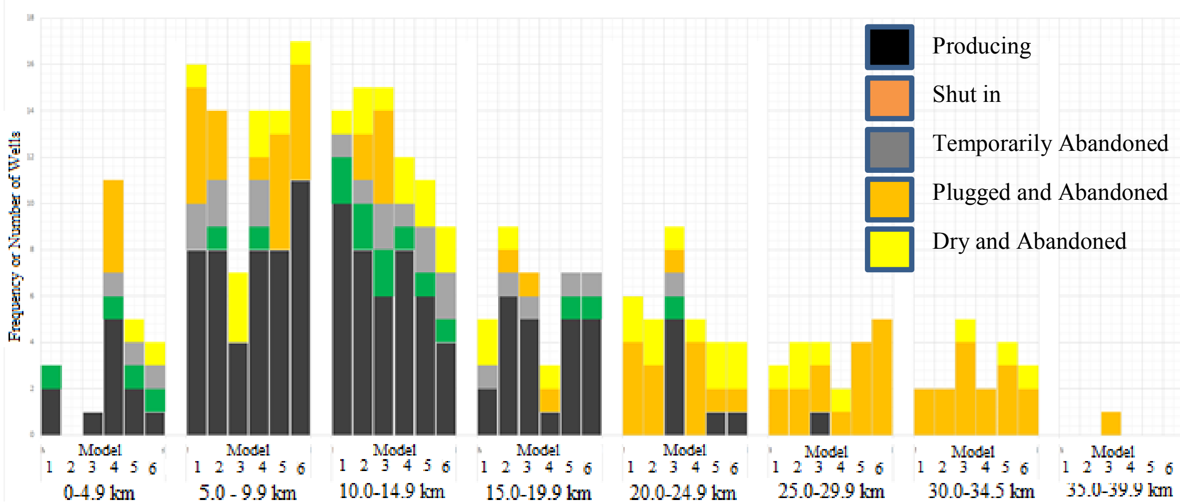
Table 5: Number of producing wells within 10 km of the epicenters of the earthquakes. The total number of producing wells in the Mobile Bay area is 22. The 68% certainty in the epicenter is in parentheses.

Earthquakes	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
August 20, 2012	10 (2.06 km)	8 (2.49 km)	5 (9.2 km)	13 (4.05 km)	10 (3.39 km)	12 (6.04 km)
August 26, 2012	13 (1.5 km)	15 (2.11 km)	8 (1.68 km)	8 (1.59 km)	15 (91.7 km)	8 (4.92 km)
September 3, 2012	7 (1.32 km)	8 (3.65 km)	0 (1.41 km)	0 (2.84 km)	0 (7.82 km)	0 (11.33 km)
November 10, 2012	15 (5.33 km)	11 (5.0 km)	8 (2.67 km)	8 (3.56 km)	16 (7.02 km)	16 (7.95 km)

Spatial Correlation of Drilled Wells with Epicenters (August 20, 2012 Earthquake)



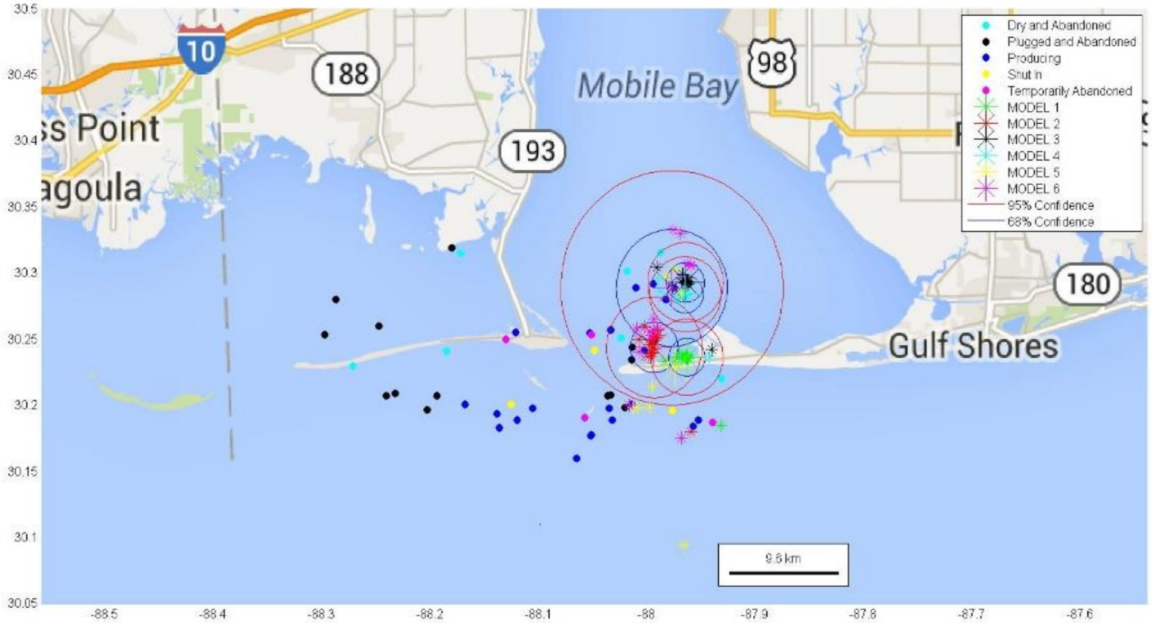
Spatial Correlation with respect to Well Status and Velocity Model
(August 20, 2012 Earthquake)



Epicentral Distance (km)

Figure 32: Spatial correlation of drilled wells around the Mobile Bay area with August 20, 2012 epicenter. (a) Map view of the correlation of the drilled wells and the epicenters from different velocity models. The blue and red lines represent the 68% and 95% confidence level in the epicenter. (b) Histogram showing the variation of the frequency of earthquakes with epicentral distance, well types and velocity models used in the absolute location determination. A spatial correlation of the drilled wells with epicentral distance shows that the producing, shut in and temporarily abandoned wells are located within 10 km of the epicenter using all the six (6) models. The plot also shows that most of the plugged and abandoned and dry and abandoned wells are located far from the epicenter which signifies that these wells are not likely to induce earthquakes.

Spatial Correlation of Drilled Wells with Epicenters (August 26, 2012 Earthquake)



Spatial Correlation with respect to Well Status and Velocity Model
(August 26, 2012 Earthquake)

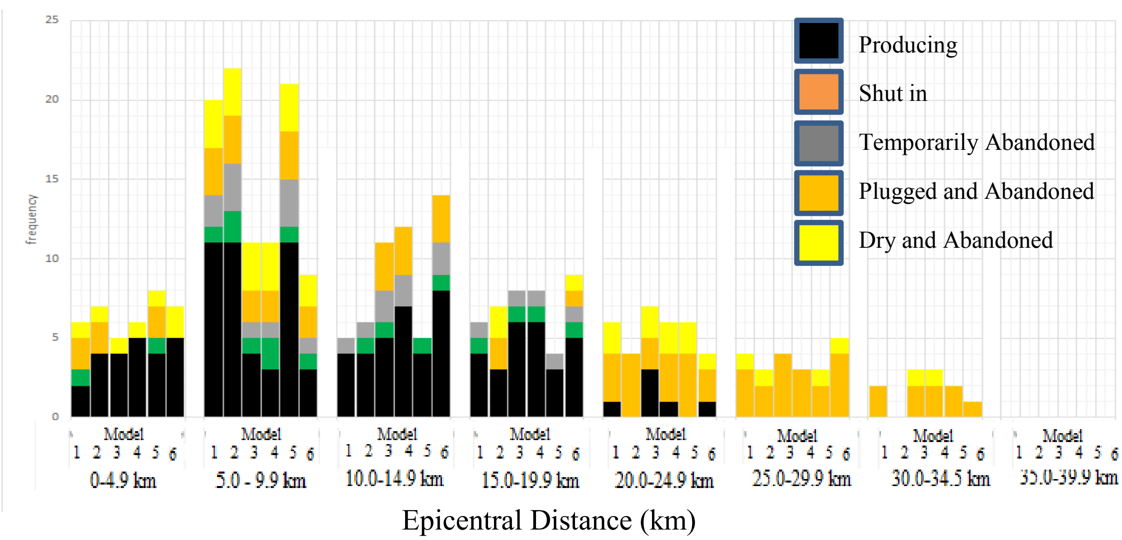
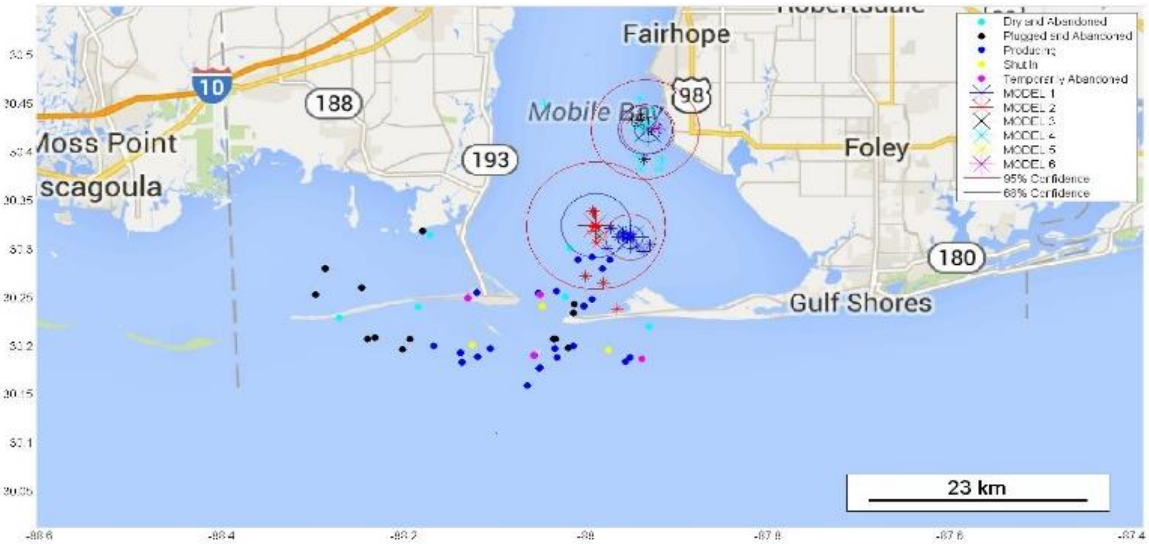
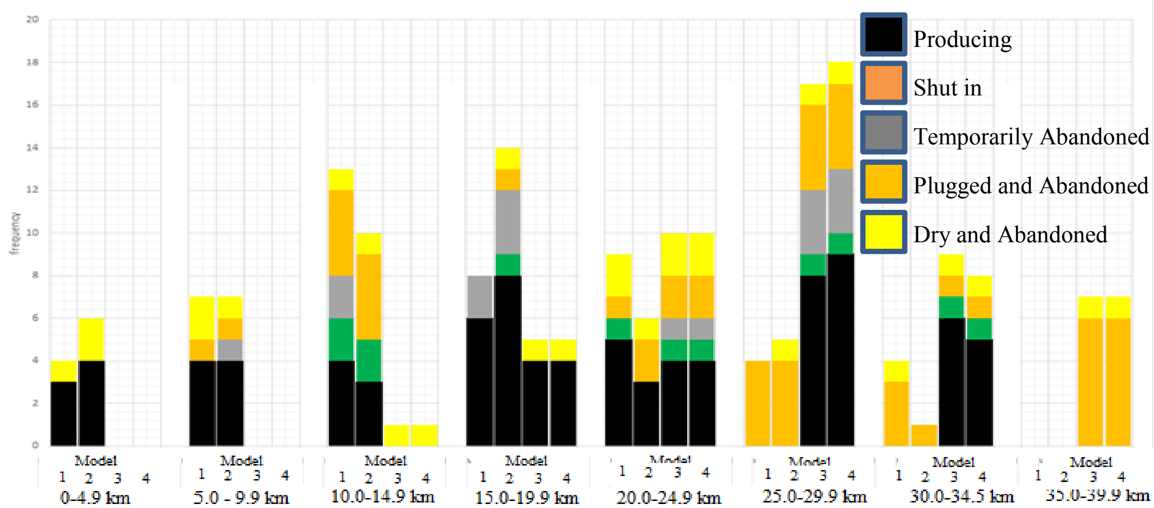


Figure 33: Spatial correlation of drilled wells around the Mobile Bay area with August 26, 2012 epicenter. (a) Map view of the correlation of the drilled wells and the epicenters from different velocity models. The blue and red lines represent the 68% and 95% confidence level in the epicenter. (b) Histogram showing the variation of the frequency of earthquakes with epicentral distance, well types and velocity models used in the absolute location determination. These observations also increase the confidence of the expected relationship between exploitation activities at these wells and the detected earthquakes.

Spatial Correlation of Drilled Wells with Epicenter (September 3, 2012 Earthquake)



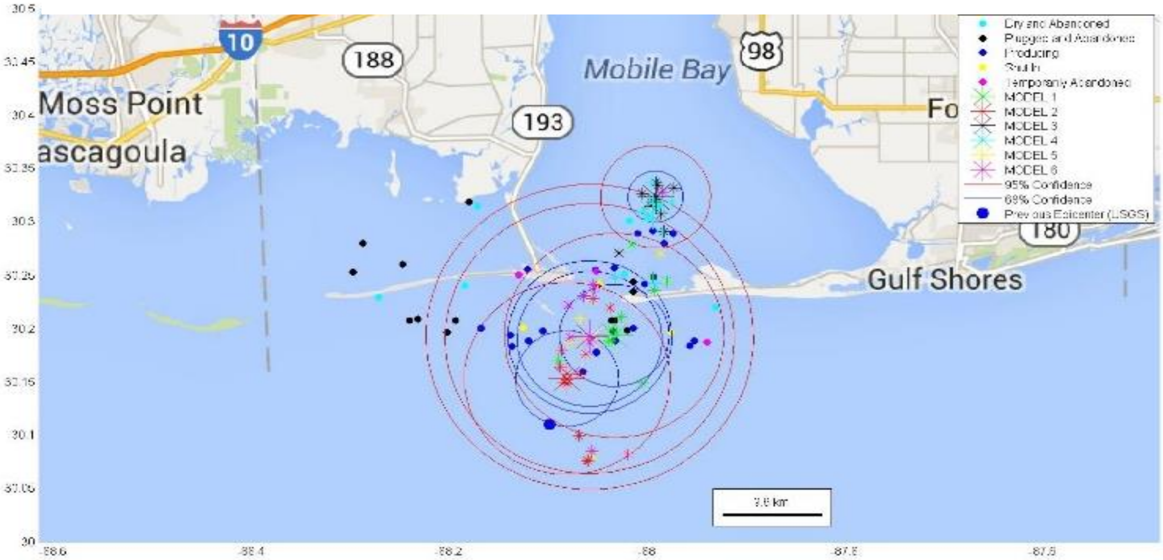
Spatial Correlation with respect to Well Status and Velocity Model
(September 3, 2012 Earthquake)



Epicentral Distance (km)

Figure 34: Spatial correlation of drilled wells around the Mobile Bay area with September 3, 2012 epicenter. (a) Map view of the correlation of the drilled wells and the epicenters from different velocity models. The blue and red lines represent the 68% and 95% confidence level in the epicenter based on the first four of the six velocity models used in this research. The other two have epicenters that are far from the expected location based on the waveform signatures and observed moveout of the P and S wave arrival times at all the TA stations. (b) Histogram showing the variation of the frequency of earthquakes with epicentral distance, well types and velocity models used in the absolute location determination. Three producing wells and one plugged and abandoned wells are within a 5 km radius of the epicenter of the earthquake using Model 1.

Spatial Correlation of Drilled Wells with Epicenter (November 10, 2012 Earthquake)



Spatial Correlation with respect to Well Status and Velocity Model
(November 10, 2012 Earthquake)

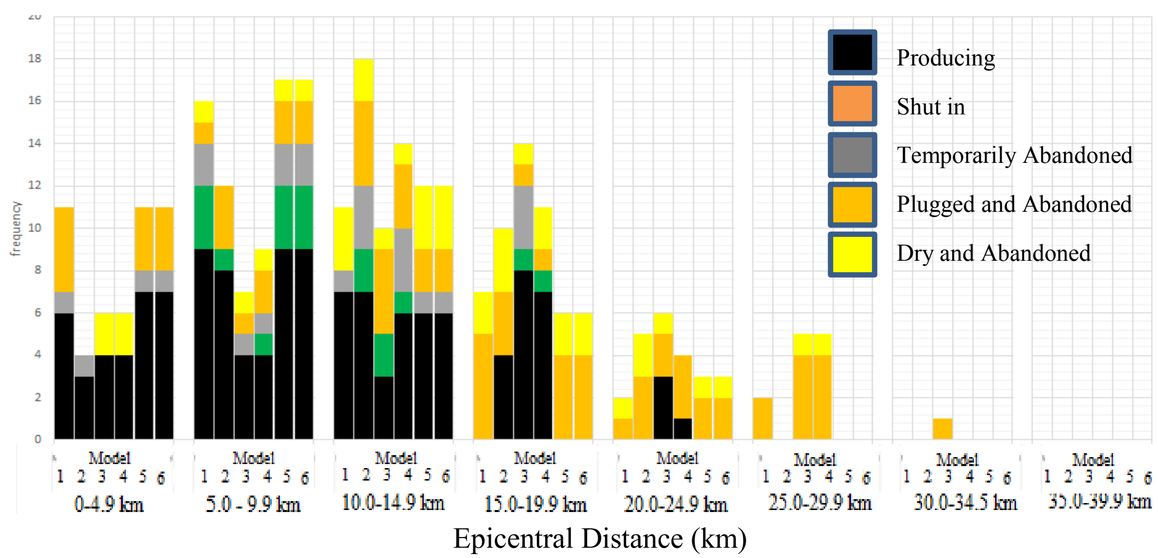


Figure 35: Spatial correlation of drilled wells around the Mobile Bay area with November 10, 2012 epicenter. (a) Map view of the correlation of the drilled wells and the epicenters from different velocity models. The blue and red lines represent the 68% and 95% confidence level in the epicenter. (b) Histogram showing the variation of the frequency of earthquakes with epicentral distance, well types and velocity models used in the absolute location determination. The histogram shows that most of the producing wells are located within 10 km using most of the models. This plot also shows that the detected earthquakes are probably due to the gas exploitation at the producing wells.

4.2 Missing earthquakes and the Gutenberg-Richter law

The Gutenberg-Richter law states that the logarithm of the frequency of earthquakes in a region has a linear relationship with the magnitude of the earthquakes. The logarithm of the frequencies was plotted against magnitudes of earthquakes in the NGM from 1974 to 2013. The plot shows that the number earthquakes with magnitudes less than 3.0 deviated negatively from the expected linear relationship (Fig. 36). This could either be due to inactivity of the region at smaller magnitudes within this time period or inefficiency in the detection of the small shallow earthquakes. Furthermore, given the amount of gas extraction at the various oil-well sites in the Mobile Bay, one would expect more small offshore earthquakes to have occurred than are known for the past 39 years.

4.3 The Poroelastic Model

One of the models that could explain the existence of the detected earthquakes in the NGM is the poroelastic model. The poroelastic model is a mathematical model that describes the expected structural deformation due to changes in pore pressure as a result of oil and gas extraction from a reservoir (Segall, 1985, Segall, 1989, Segall, 1992, Segall et al., 1994). The model predicts the horizontal and vertical displacements and horizontal strain as a function of distance across the surface extent of the reservoir.

The extraction of oil and gas, based on the poroelastic model, has two opposing effects on fault stability. On one hand, extraction leads to a decrease in pore fluid pressure, which increases the effective mean stress and should thus decrease the

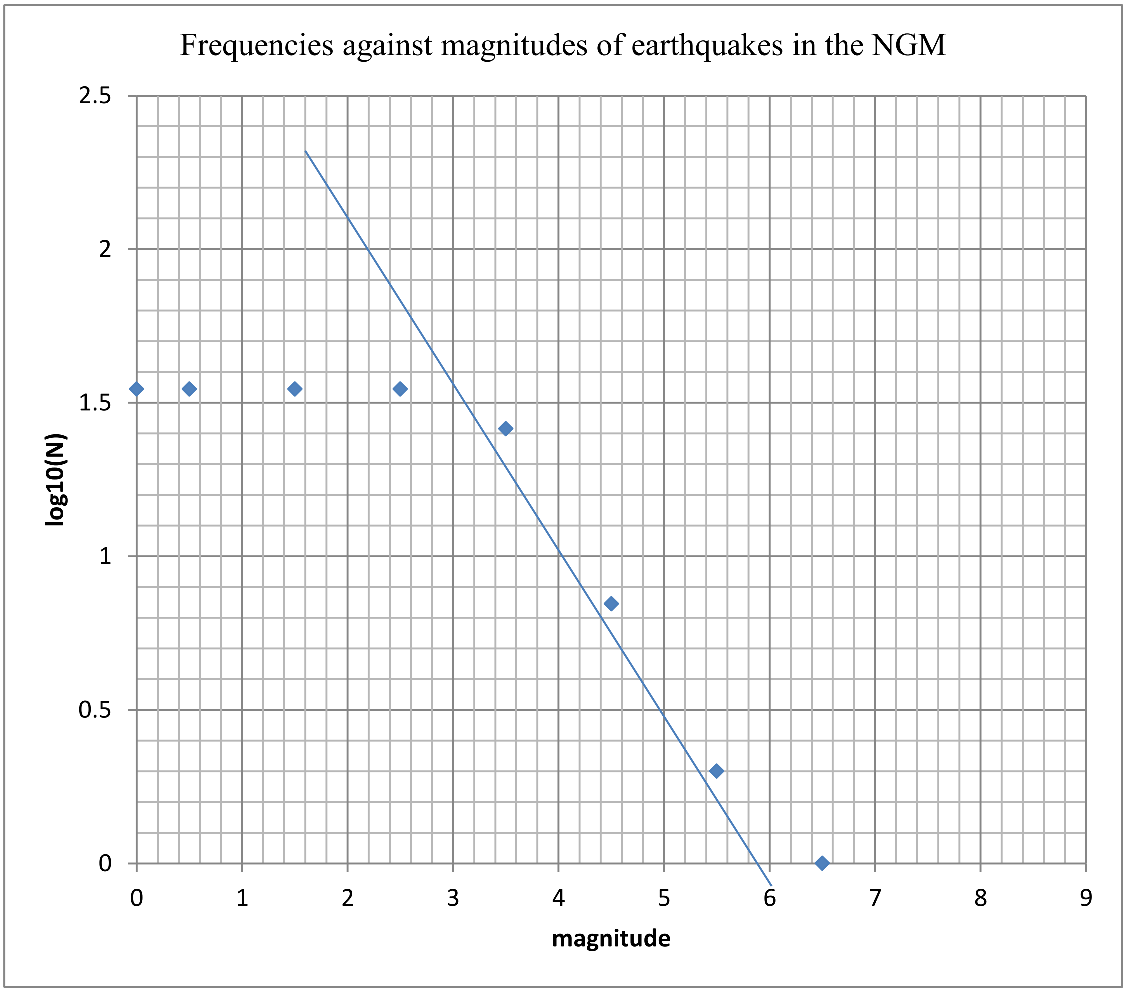


Figure 36: The plot of the logarithm of the number of earthquakes above certain magnitudes of earthquakes in the NGM from 1974 to 2013 against magnitude. The blue points represent the expected relationship based on the Gutenberg-Richter law. The plot shows that the number earthquakes with magnitudes below 3 deviates negatively from the expected Gutenberg-Richter linear relationship.

likelihood of a fault to slip. On the other hand, differential compaction resulting from the large volume of oil and gas extracted from the well increases the probability of subsidence and an increase in horizontal strains (Segall, 1985; Segall et al., 1994). The effect of this differential compaction is expected to dominate the “locking” effect on a fault due to reduction in the pore pressure, leading to an increase in the differential stress. This differential stress results in failure of a pre-existing fault, which happens when the frictional resistance is overcame. The increasing horizontal strains causes reverse faults in the rocks above and below the reservoir, whereas the rock on the flanks of the field is extended, leading to normal faults (Segall, 1985 in Odonne, 1999, Fig. 37a). Bardainne et al. (2008) argued that even though a field is strained due to oil and gas extraction, induced earthquakes are only observed where pre-existing faults are parallel to the isobaths of the reservoir roof and where the faults plunge towards field’s flank.

Examples of oil and gas fields where the observed local earthquakes were attributed to the oil and gas extraction are the Lacq field, France, the Wilmington field, USA and the Strachan field, Canada. The spatial distribution of epicenters of the detected earthquakes within Lacq oil and gas fields is comparable to the aerial extent of gas reservoir than the oil field, suggesting a high susceptibility of gas fields to trigger earthquakes when certain thresholds are exceeded (Segall et al., 1994; Baronova et al., 1999). The gas pressure in the Lacq field is also observed to be linearly proportional to the surface subsidence (Segall et al., 1994) with maximum surface subsidence at the center of the field (Segall, 1989). This observed subsidence is also in accordance with the horizontal and displacement at the Wilmington field, California (Yerkes and Castles, 1970). Furthermore, the detected earthquakes within the Strachan field, Canada have a

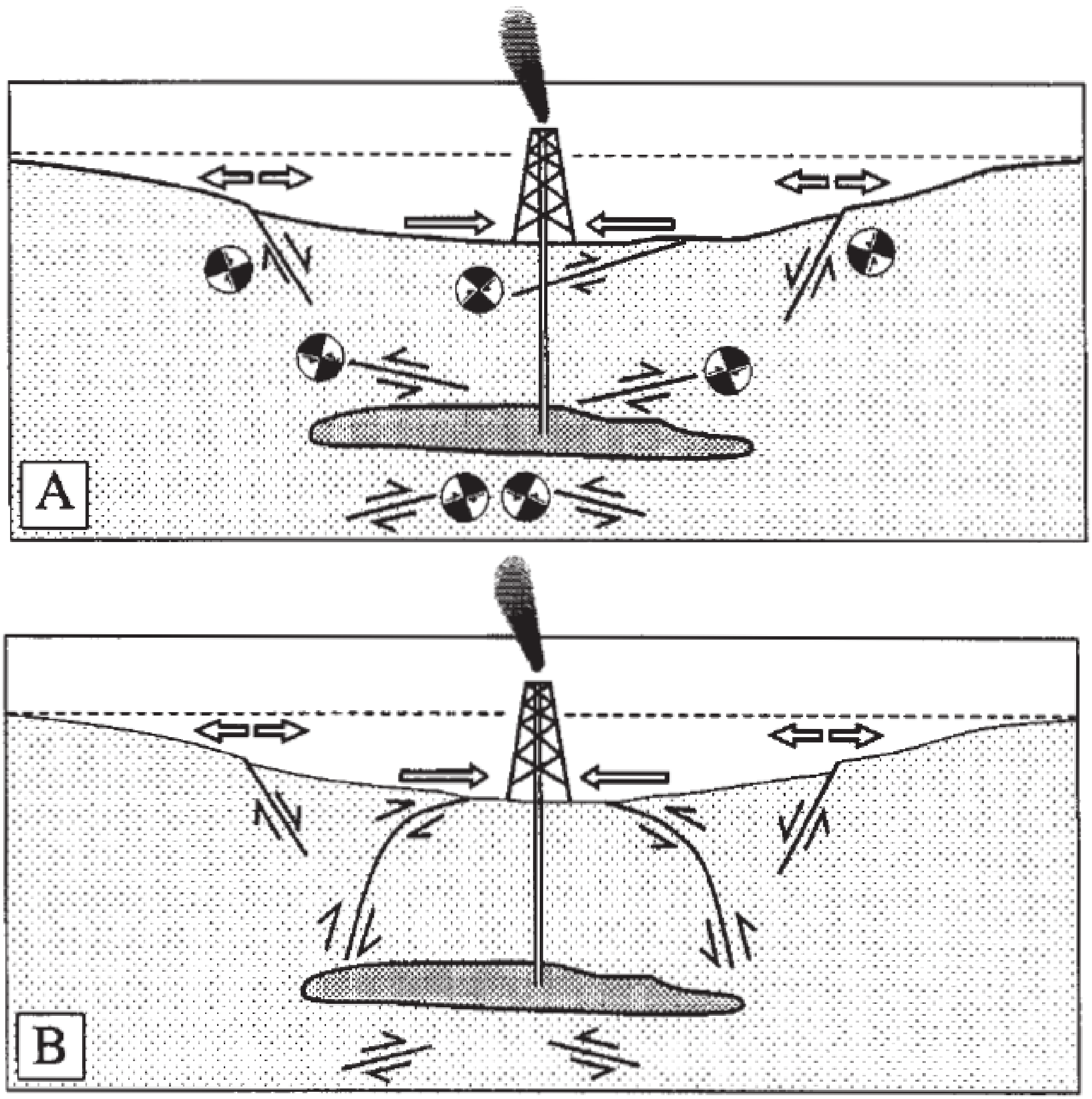


Figure 37: A schematic cross section of the poroelastic model of Segall (1989) showing the surface deformation, faulting, and simulated focal mechanism associated with oil and gas extraction. (a) Reversed faults were proposed to be above and below the reservoir while normal faults at either flank of the field. The simulated focal mechanisms from the model are also shown. Surface horizontal displacement are represented as the white arrows (Segall 1989 in Odonne, 1999) (b) Modified poroelastic model using analog experiments with high angle reversed faults above the reservoir in contrast to the low angle proposed by Segall (1989) while other predictions remain the same. The high angle faults are favored because the other nodal planes were chosen to be the fault planes above the reservoir forming a cone shape.

strong direct correlation with the rates of production and long history of gas extraction in the field, and the earthquakes are mostly concentrated near the gas producing wells in the field and are far less abundant near oil-producing areas (Baronova et al., 1999).

A modified version of the poroelastic model using a deflated-reservoir sand-silicone analog experiment with the reservoir as latex balloon or undercompacted ground sand show different results from the mathematical model (Odonne, 1999).The reservoirs were overlain by dry noncohesive sand from Fontainebleau (Paris basin, France) with a coefficient of friction, h, of 0.60 and can represent the brittle behavior of the upper crust (Odonne, 1999). The reservoirs were depleted by deflating the latex balloon and pump air from the confined undercompacted ground sand volume and the attitude of the generated faults within the overlain noncohesive sand were observed (Odonne, 1999). Odonne (1999) observed high and low angle reverse faults at the top of and below the reservoir, respectively, in contrast with a low angle reverse fault of the poroelastic model at both the top of and below the reservoir. These faults produce surface central subsidence with respect to distance across the reservoir and are consistent with the observed subsidence at the Lacq field (Odonne, 1999, Fig. 37b).

CHAPTER FIVE: CONCLUSION

The formulation of the earthquake detection algorithm used in this study is very sensitive to detect small shallow earthquakes. It uses the stack of seismic traces from several TA stations. The detection of five (5) new earthquakes that had not been detected by the conventional seismic monitoring systems from 2011 to 2013 in the NGM is the main contribution of the research. Three of the newly detected earthquakes are in the gas field in Mobile Bay. The spatial correlation of the earthquakes with the producing wells in this area can be explained by the poroelastic model of induced earthquakes due to oil and gas extraction (Segall, 1985; Segall, 1989; Segall, 1992; and Segall et al., 1994).

The research has a few limitations. Only four earthquakes have occurred in the Mobile Bay area, and they appear to be correlated with gas extraction. The small sample size is a limitation in attributing the observed earthquakes to the gas extraction in the field. The focal mechanisms (strike, dip and other geometrical properties) of the detected earthquakes are not possible to be determined due to lack of sufficient areal and azimuthal distribution of the TA stations around the epicenters. Knowledge about the causative fault orientations would have aided the correlation of earthquakes distribution with the oil and gas extraction. The orientation of the fault geometry is also very important. Even though the entire field is strained due to the gas extraction, the induced seismicity can only occur where pre-existing faults are perpendicular to the direction of pressure changes within the reservoir and where faults dip towards boundary of the field (Segall, 1989; Bardainne et al., 2008). Proprietary information in the gas wells such as specific starting times, duration and rate of gas extraction, if available, would have permitted a temporal correlation of the detected earthquakes with the gas extraction. Induced earthquakes are expected to occur after the gas extraction. The time lapse between the oil and gas extraction and induced earthquakes is also data that could help to further understand the mechanism of induced earthquakes due to oil and gas extraction.

In conclusion, the detected events in Mobile Bay may be related to or induced by the extraction of gas from the producing wells because 45.5%, 59.1%, 31.8% and 68.2% of the producing wells are within 10 km of the epicenters of the August 20, 2012, August 26, 2012, September 3, 2012 and November 10, 2012 earthquakes, respectively. Most of the 'plugged and abandoned' wells and 'dry and abandoned' wells are evenly distributed in the Mobile Bay area and hence likely have no contribution to the induced earthquakes.

I recommend that the government and/or other private organizations site seismometers in the offshore region of the NGM to ensure full azimuth coverage of the stations for focal mechanism determination, proximity to the epicenter, and effective determination of focal depth for proper correlation of focal depth with reservoir depths. The offshore seismometers will also allow more local earthquakes, if present, to be detected, and hence increase the number of earthquakes correlated with oil and gas extraction in the NGM.

Reference

Bardainne, T.; Dubos-Sallee, N., Senechal, G., Gaillot. P. and H. Perroud (2008). Analysis of the induced seismicity of the Lacq gas field (Southwestern France) and model of deformation. Geophys. J. Int., vol. 172, pp. 1151-1162.

Baronova, V., Mustaqeem, A. and S. Bell (1999). A model for induced seismicity caused by hydrocarbon production in the Western Canada sedimentary basin. Can. J. Earth Sci. vol. 36, pp. 47-64.

Bureau of Safety and Environemntal Enforcement (BSEE), 2014 [Internet]. Available from http://www.data.bsee.gov/homepg/data_center/well/war/warlist.asp

Chapman, M. and S. Pezeshk (2014). 2014 CEUS NEHRP Research review and planning workshop, Memphis, TN.

Chulick, G. and W. Mooney (2002) Seismic Structure of the Crust and Uppermost Mantle of North America and Adjacent Oceanic Basins: A Synthesis. Bulletin of the Seismological Society of America, Vol. 92, No. 6, pp. 2478–2492.

Ebel, J. (1994). The Mlg (f) magnitude scale: A proposal for its use for its use for northeastern North America. Seismological Research Letters, vol. 65, no. 2, pp. 157-166.

Gendron, P., Ebel, J.E. and D. Manolakis (2000). Rapid Joint Detection and Classification with Wavelet Bases via Bayes Theorem. Bulletin of the Seismological Society of America, Vol. 90, No. 3, pp. 764-774.

Holt J., Bartz M. and J. Lehman (1983). Final Regional Environemntal Impact Statement Volume 1. Gulf of Mexico. Minerals Management Service Gulf of Mexico OCS Region.

Incorporated Research Institutions for Seismology (IRIS) Earthquake Browser [Internet]. [updated Oct. 15, 2013] Available from http://www.iris.edu/ieb/index.html?format=text&nodata=404&starttime=1970-01-01&endtime=2025-01-01&minmag=0&maxmag=10&mindepth=0&maxdepth=900&orderby=time-desc&limit=200&maxlat=30.87&minlat=22.51&maxlon=-85.36&minlon=-96.97

Jonathan Sulian [Internet] (2011). Automatic map scale generation. Available from http://www.mathworks.com/matlabcentral/fileexchange/33545-automatic-map-scale-generation/content/makescale.m

Keranen, K., Savage, H., Abers, G. and E. Cochran (2013). Potentially induced earthquakes in Oklahoma, USA: Links between wastewater injection and the 2011 M_w 5.7 earthquake sequence. Geology, vol. 41, no. 6, pp. 699-702.

Klein, F. W. (2002). User’s guide to Hypoinverse-2000, a FORTRAN program to solve for earthquake locations and magnitude, U.S. Geol. Surv. Open-File Rep. 02–171, 121.

Maury, V., Grasso, J. and G. Wittlinger (1992). Monitoring of subsidence and induced seismicity in the Lacq Gas Field (France): the consequences on gas production and field operation. Engineering geology, vol. 32, pp. 123-135.

Meschede M. and W. Frisch (1998). A plate-tectonic model for the Mesozoic and Early Cenozoic history of the Caribbean plate Tectonophysics 296, 269–291.

Mousa W.A. and Al-Shuhail A.A. (2009). Processing of Seismic Reflection Data using MATLAB.

Odonne F., Menard I., Massonnat G. and J. Rolando (1999). Abnormal reverse faulting above a depleting reservoir. Geology vol. 27, no. 2, pp. 111-114.

Schilling R and S. Harris (2012). Fundamentals of Digital Signal Processing using MATLAB, 2nd Edition.

Segall, P. (1985). Stress and subsidence resulting from subsurface fluid withdrawal in the epicentral region of the 1983 Coalinga Earthquake. Journal of Geophysical Research, vol. 90, no. B8, pp. 6801-6816.

Segall, P. (1989). Earthquakes triggered by fluid extraction. Geology, vol. 17, pp. 942-946.

Segall, P., Grasso J., and A. Mossop (1994). Poroelastic stressing and induced seismicity near the Lacq gas field, southwestern France. Journal of Geophysical Research, vol. 99, no. B8, pp. 15,423-15,438.

United State Energy Information Administration [Internet]. [Release date: 9/27/2013] Available from http://www.eia.gov/dnav/pet/pet_crd_crpdn_adc_mbbl_a.htm

Yerkes R. and R. Castle (1976). Seismicity and faulting attributable to fluid extraction, Eng. Geol., vol. 10, pp. 151-167.

Yilmax O. (2001). Seismic Data Analysis: Processing, Inversion and Interpretation of seismic data: 2nd (second) edition.

Zohar Bar-Yehuda [Internet] (2014). Zoharby/plot_google_map. MATLAB File exchange. Available from http://www.mathworks.com/matlabcentral/fileexchange/27627-zoharby-plot-google-map

Appendix A

Transportable Array Stations

Mississippi 2011.txt

State Code	Station Code	Network	Starting time	Ending time	Latitude	Longitude
MS	TA	247A	4/9/2011	1/21/2013	32.1	-88.6
MS	TA	144A	3/22/2011	2/1/2013	32.6	-90.4
MS	TA	145A	3/25/2011	3/16/2013	32.6	-89.9
MS	TA	146A	3/23/2011	1/29/2013	32.6	-89.1
AL	TA	147A	4/10/2011	3/18/2013	32.7	-88.3
MS	TA	245A	3/20/2011	1/20/2013	32	-89.9
MS	TA	246A	3/26/2011	2/8/2013	32	-89.1
MS	TA	244A	3/19/2011	1/22/2013	32	-90.7
LA	TA	545A	3/11/2011	12/9/2012	30	-90.5
LA	TA	645A	3/13/2011	12/9/2012	29.5	-90.6
MS	TA	446A	3/15/2011	12/13/2012	30.8	-89.4
LA	TA	546A	3/16/2011	12/12/2012	30.2	-89.7
MS	TA	344A	3/17/2011	12/14/2012	31.5	-90.7
LA	TA	646A	3/17/2011	8/29/2012	29.6	-89.8
MS	TA	345A	3/18/2011	12/14/2012	31.3	-90
MS	TA	346A	3/18/2011	2/5/2013	31.4	-89.5
MS	TA	447A	3/18/2011	12/13/2012	30.8	-88.7
AL	TA	347A	3/19/2011	12/15/2012	31.4	-88.5
AL	TA	348A	3/20/2011	1/25/2013	31.4	-87.9
AL	TA	448A	4/7/2011	3/18/2013	30.9	-87.9
AL	TA	Y47A	4/13/2011	2/3/2013	33.9	-87.8
MS	TA	Y46A	4/14/2011	2/6/2013	33.9	-88.9
AL	TA	Z48A	4/15/2011	2/3/2013	33.4	-87.6
MS	TA	Y45A	4/16/2011	2/6/2013	33.9	-89.5
MS	TA	Y44A	4/16/2011	2/8/2013	34	-90.2
AL	TA	Z47A	4/16/2011	1/31/2013	33.2	-88.1
MS	TA	Z46A	4/17/2011	1/29/2013	33.2	-88.9
MS	TA	Z44A	4/18/2011	1/17/2013	33.3	-90.4
MS	TA	Z45A	4/19/2011	2/4/2013	33.4	-89.7
MS	TA	X44A	5/11/2011	2/8/2013	34.5	-90.1
MS	TA	X45A	5/17/2011	2/7/2013	34.4	-89.4
LA	TA	241A	2/10/2011	12/1/2012	32	-92.9
LA	TA	240A	2/11/2011	12/1/2012	32	-93.8
LA	TA	243A	2/19/2011	11/30/2012	31.9	-91.5

LA	TA	242A	2/21/2011	11/13/2012	32.1	-92.2
LA	TA	342A	2/23/2011	12/2/2012	31.4	-92.3
LA	TA	442A	2/24/2011	11/28/2012	30.7	-92.4
LA	TA	443A	2/24/2011	11/28/2012	30.8	-91.8
LA	TA	343A	2/24/2011	11/30/2012	31.3	-91.6
LA	TA	444A	3/10/2011	12/8/2012	30.7	-90.7
LA	TA	445A	3/10/2011	12/8/2012	30.7	-90.3
LA	TA	541A	2/26/2011	12/6/2012	30.1	-93.2
LA	TA	544A	2/24/2011	2/2/2013	30.1	-91.2
LA	TA	341A	3/8/2011	11/15/2012	31.3	-93.2
AL	TA	248A	4/8/2011	1/25/2013	32.1	-87.7
MS	TA	Y43A	4/13/2011	2/24/2013	33.9	-90.9
LA	TA	140A	2/12/2011	11/10/2012	32.6	-93.6
LA	TA	141A	2/14/2011	11/12/2012	32.6	-92.9
LA	TA	142A	2/22/2011	11/13/2012	32.5	-91.9
LA	TA	143A	2/18/2011	12/10/2012	32.7	-91.4
LA	TA	441A	2/24/2011	12/6/2012	30.8	-93.2
LA	TA	542A	2/24/2011	12/7/2012	30.1	-92.6

Mississippi 2012.txt

State Code	Station Code	Network	Starting time	Ending time	Latitude	Longitude
AL	US	BRAL	8/14/2006	3/18/2013	31.2	-87.1
LA	TA	445A	3/10/2011	12/8/2012	30.7	-90.3
MS	TA	446A	3/15/2011	12/13/2012	30.8	-89.4
MS	TA	345A	3/18/2011	12/14/2012	31.3	-90
MS	TA	346A	3/18/2011	2/5/2013	31.4	-89.5
MS	TA	447A	3/18/2011	12/13/2012	30.8	-88.7
MS	TA	244A	3/19/2011	1/22/2013	32	-90.7
AL	TA	347A	3/19/2011	12/15/2012	31.4	-88.5
MS	TA	245A	3/20/2011	1/20/2013	32	-89.9
AL	TA	348A	3/20/2011	1/25/2013	31.4	-87.9
MS	TA	146A	3/23/2011	1/29/2013	32.6	-89.1

AL	TA	448A	4/7/2011	3/18/2013	30.9	-87.9
AL	TA	248A	4/8/2011	1/25/2013	32.1	-87.7
MS	TA	247A	4/9/2011	1/21/2013	32.1	-88.6
AL	TA	147A	4/10/2011	3/18/2013	32.7	-88.3
AL	TA	Y47A	4/13/2011	2/3/2013	33.9	-87.8
MS	TA	Y46A	4/14/2011	2/6/2013	33.9	-88.9
LA	TA	Z48A	4/15/2011	2/3/2013	33.4	-87.6
MS	TA	Y45A	4/16/2011	2/6/2013	33.9	-89.5
AL	TA	Z47A	4/16/2011	1/31/2013	33.2	-88.1
MS	TA	Z46A	4/17/2011	1/29/2013	33.2	-88.9
MS	TA	X45A	5/17/2011	2/7/2013	34.4	-89.4
AL	TA	X47A	12/10/2011	3/18/2013	34.5	-87.9
AL	TA	X49A	12/11/2011	3/18/2013	34.5	-86.3
MS	TA	X46A	12/11/2011	3/18/2013	34.6	-88.6
AL	TA	X48A	12/14/2011	3/18/2013	34.5	-87
AL	TA	Y48A	12/18/2011	3/18/2013	33.9	-87.2
AL	TA	Y49A	12/18/2011	3/18/2013	33.9	-86.4
AL	TA	Z49A	12/19/2011	3/18/2013	33.2	-86.5
AL	TA	Y50A	12/21/2011	3/18/2013	33.9	-85.7
AL	TA	Z50A	12/26/2011	3/18/2013	33.3	-85.9
AL	TA	148A	1/16/2012	2/19/2013	32.6	-87.6
AL	TA	149A	1/17/2012	3/18/2013	32.6	-86.8
FL	TA	449A	1/18/2012	3/18/2013	30.8	-87.2
AL	TA	150A	1/18/2012	3/18/2013	32.6	-86
FL	TA	450A	1/19/2012	3/18/2013	30.8	-86.6
AL	TA	349A	1/20/2012	3/18/2013	31.4	-87.2
AL	TA	249A	1/21/2012	3/18/2013	32	-87.1

AL	TA	250A	1/24/2012	3/18/2013	32	-86.3
AL	TA	251A	1/24/2012	3/18/2013	32.1	-85.4
AL	TA	351A	2/2/2012	3/18/2013	31.3	-85.6
FL	TA	451A	2/3/2012	3/18/2013	30.6	-85.7
AL	TA	X50B	4/4/2012	3/18/2013	34.5	-85.7

Mississippi 2013.txt

State Code	Station Code	Network	Starting time	Ending time	Latitude	Longitude
AL	US	BRAL	8/14/2006	3/18/2013	31.2	-87.1
AL	TA	151A	1/21/2012	3/18/2013	32.5	-85.3
AL	TA	350A	1/21/2012	3/18/2013	31.4	-86.3
MS	TA	145A	3/25/2011	3/16/2013	32.6	-89.9
AL	US	LRAL	7/2/2001	3/18/2013	33	-87
GA	TA	352A	3/7/2012	3/18/2013	31.5	-84.9
GA	TA	353A	3/9/2012	3/18/2013	31.3	-84.2
GA	TA	253A	3/11/2012	3/18/2013	32.1	-84.1
GA	TA	254A	3/14/2012	3/18/2013	31.9	-83.3
AL	TA	448A	4/7/2011	3/18/2013	30.9	-87.9
GA	TA	252A	3/11/2012	3/18/2013	32	-84.7
AL	TA	147A	4/10/2011	3/18/2013	32.7	-88.3
GA	TA	153A	3/19/2012	3/18/2013	32.7	-83.8
GA	TA	154A	3/20/2012	3/18/2013	32.6	-83.1
GA	TA	152A	4/3/2012	3/18/2013	32.7	-84.7
AL	TA	X47A	12/10/2011	3/18/2013	34.5	-87.9
AL	TA	X49A	12/11/2011	3/18/2013	34.5	-86.3
MS	TA	X46A	12/11/2011	3/18/2013	34.6	-88.6

AL	TA	X48A	12/14/2011	3/18/2013	34.5	-87
AL	TA	Y48A	12/18/2011	3/18/2013	33.9	-87.2
AL	TA	Y49A	12/18/2011	3/18/2013	33.9	-86.4
AL	TA	Z49A	12/19/2011	3/18/2013	33.2	-86.5
AL	TA	Y50A	12/21/2011	3/18/2013	33.9	-85.7
AL	TA	Z50A	12/26/2011	3/18/2013	33.3	-85.9
GA	TA	256A	3/19/2012	3/18/2013	32	-81.9
AL	TA	149A	1/17/2012	3/18/2013	32.6	-86.8
FL	TA	449A	1/18/2012	3/18/2013	30.8	-87.2
AL	TA	150A	1/18/2012	3/18/2013	32.6	-86
FL	TA	450A	1/19/2012	3/18/2013	30.8	-86.6
AL	TA	349A	1/20/2012	3/18/2013	31.4	-87.2
AL	TA	249A	1/21/2012	3/18/2013	32	-87.1
AL	TA	250A	1/24/2012	3/18/2013	32	-86.3
AL	TA	251A	1/24/2012	3/18/2013	32.1	-85.4
AL	TA	351A	2/2/2012	3/18/2013	31.3	-85.6
FL	TA	451A	2/3/2012	3/18/2013	30.6	-85.7
AL	TA	X50B	4/4/2012	3/18/2013	34.5	-85.7
FL	TA	452A	2/5/2012	3/18/2013	30.8	-85.2
FL	TA	552A	2/6/2012	3/18/2013	30.1	-85.3
FL	TA	553A	2/7/2012	3/18/2013	30.2	-84.4
GA	TA	453A	2/9/2012	3/18/2013	30.9	-84.3
GA	TA	455A	2/12/2012	3/18/2013	30.7	-83
GA	TA	454A	2/13/2012	3/18/2013	30.7	-83.6
FL	TA	554A	2/13/2012	3/18/2013	30.1	-83.7
FL	TA	555A	2/14/2012	3/18/2013	30.1	-83

Appendix B
Event Extraction Algorithm

```
------------------------------------------------------
clear all; clc; close all;

%  Summary of the Processing Sequence
%  TA waveform records were retrieved from the IRIS database and a narrow bandpass
%  filter of 1-2 Hz was applied to remove background and high frequency signals and
%  noise. A prestack RMS Automatic Gain Control (AGC) was applied to boost the
%  signals at further stations. Previous earthquakes were analyzed in order to determine
% the moveout velocity of the P- and S- waves due to different arrival times as they
% propagate across the TA network.
%       The Hilbert transform was used to determine waveform envelopes to avoid
% destructive stacking. A weighted stacking method was applied separately to the P and
% S waves. An event was detected when the P- and S- SNRs are greater than 10. The
% seismic records were plotted with respect to station distance from the Gulf for better
% visualization and discrimination of events.

%       This program requires the following MATLAB functions to run:
%       1. haversine.m  to determine distance between two points on a sphere (earth
%           surface) (written by Josiah Renfree, 2010)
%       2. importfile.m to importing the text files containing the TA stations.
%       3. ReadSacBinary.m to read in the seismic data (SAC format) into Matlab (written
%           by Prof John Ebel)
%       4. section_display.m  to display the waveform data as a function of epicentral
%          distance (modified after Wail A. Mousa and Abdullatif A. Al-Shuhail)

%       Input:
%       1. Seismic data in SAC format
%       2. Text file containing the used TA stations, and in the format:
%       state code |station code | Network | starting time | ending time | latitude | longitude
%       e.g. AL  US  BRAL    8/14/2006   3/18/2013   31.2    -87.1

%       Output:
%       1.  Seismic section of the detected event (Seismic Section_1_38ST_10-16_3_13_
%           30.1- 88.pdf)
%       2.  Peak amplitude of the time window (Peak Amplitude of Envelopes_10-16_3_
%            13_30.1-88.1.txt)
%       3.  Start and end time of the time window where an earthquake was detected
%           (startTimes_endTime_10-16_3_13_30.1-88.1.txt)
%       4.  Standard deviation of the background noise, P- and S-waves (STD of Events
%            with Stations_10-16_3_13_30.1-88.1.txt)
%       5.  SNR_P_S and STD_P_S_NOI_10-16_3_13_30.1-88.1.txt

%       By Oluwaseun I. Fadugba and John E. Ebel (Project Supervisor) - July 2015
```

```
%       Department of Earth and Environmental Sciences, Boston College, MA
%       This code is an integral part of an MSc Thesis titled "Detection of Induced
%       Seismicity due to Oil and Gas Extraction in the Northern Gulf of Mexico, USA

lat1=30.1; long1=-88.1; % Epicenter of the November 10, 2012 earthquakes. This is was
% used as the reference for the section_display.m function

freq1= 1.0; freq2= 2.0; % Butterworth Bandpass filter frequency ranges (freq1 to freq2)

tic  % tic determines the time it takes the code to be completed

%       State the start and end times for the first data window in the TA station data
%       Format: 'yyyy-mm-dd hh:mm:ss'
startTime='2013-01-06 00:00:00'; % start time of the first data window
endTime='2013-01-06 00:04:00'; % end time of the first data window
file_tag=sprintf('%s_%2.1f%2.1f','14-18_11_12',lat1,long1);
%       a file tag for November 14-18, 2012 seismic data e.g. 14-18_11_12_30.1-88.1

%  concatenating the filenames of the seismic data to be in the same format with the
%  filename from the conversion of SEED to SAC format
Stat11 ='2013.006.00.00.00.0000' ;
Stat12 ='2013.007.00.00.00.0000' ; % this is necessary because the Jrdseed sometimes
% breaks the SEED data into more than one SAC file.
Stat13 ='2013.008.00.00.00.0000' ; % same as above

Minutes_to_Analyze=13000; % length of TA data in the file to be processed in minutes
startTime0= datenum(startTime)+(2/24/60);  % start time of the first data window with 2
%  minute increment. The 2/24/60 represents 2 minutes in days.
endTime0= datenum(endTime)+(2/24/60);  % same as above

z=file_tag;

[lat_long, Stat_Net_Sta]=importfile ('Mississippi 2013.txt'); % the TA stations

quest=0; % Zero (0) if an earthquake has not been detected in the previous data window
% the value will change to 1 if an earthquake has just been detected, skip the 4-minute
% data and then use the previous 4-8 minute data window as the background noise.

loc1= [lat1 long1]; %the reference earthquake epicenter

time_diff=endTime0-startTime0; %time width of each data window (4 minutes).
total_time=round(time_diff*60*24)*60; % unit in seconds. The endTime - startTime will
% be in days but needed to be converted to seconds
tot=40*total_time; % Conversion of the time to number of samples. The sampling rate
% was 40 samples per second
```

```
 m=0.5*(Minutes_to_Analyze-round(time_diff*60*24))+1; % determining the number of
%       intervals needed. Note: there are 4 minute data windows each with 2-minute
overlap.

k=0; i=1;
%%      Setting up filenames for the outputs.
file1='Peak Amplitude of Envelopes'; % peak amplitudes of envelope of the data window
% where an earthquake is detected.
file3='STD of Events with Stations'; % standard deviation of the amplitudes
file2='SNR_P_S and STD_P_S_NOI'; % signal-to-noise ratio of the P- and S-waves
file4='startTimes_endTime'; % start and end time of the detected earthquakes
ext='.txt'; % extensions for the text files
ext2='.SAC'; % extension for the input seismic data files (SAC format)

% concatenating the filenames e.g. "startTimes_endTime.txt"
filename1 = sprintf('%s_%s%s',file1,z,ext);
filename3 = sprintf('%s_%s%s',file3,z,ext);
filename2 = sprintf('%s_%s%s',file2,z,ext);
filename4 = sprintf('%s_%s%s',file4,z,ext);

pdf_name = sprintf('%s_%s','Stacked Envelopes',z);

%  Output files (in postscript) that contain the detected events. The output is split into 12
%  files to avoid oversize postscript files
pdf_name21 = sprintf('%s_%s','Seismic Section_1_38ST',z);
%  e.g. Seismic Section_1_38ST_6-12_1_13_30.1-88.1
pdf_name22 = sprintf('%s_%s','Seismic Section_2_38ST',z);
pdf_name23 = sprintf('%s_%s','Seismic Section_3_38ST',z);
pdf_name24 = sprintf('%s_%s','Seismic Section_4_38ST',z);
pdf_name25 = sprintf('%s_%s','Seismic Section_5_38ST',z);
pdf_name26 = sprintf('%s_%s','Seismic Section_6_38ST',z);
pdf_name27 = sprintf('%s_%s','Seismic Section_7_38ST',z);
pdf_name28 = sprintf('%s_%s','Seismic Section_8_38ST',z);
pdf_name29 = sprintf('%s_%s','Seismic Section_9_38ST',z);
pdf_name210 = sprintf('%s_%s','Seismic Section_10_38ST',z);
pdf_name211 = sprintf('%s_%s','Seismic Section_11_38ST',z);
pdf_name212 = sprintf('%s_%s','Seismic Section_12_38ST',z);

filename1 = char(filename1); % converting the concatenated filenames to character
filename3 =char(filename3);
filename2 = char(filename2);
filename4 =char(filename4);
%
fid2 = fopen(filename2,'w');  % open a file called filename2 and enable the write mode
fclose(fid2); % close the opened file
fid3 = fopen(filename3,'w');  fclose(fid3);
```

```
fid4 = fopen(filename4,'w'); fclose(fid4);

%%     Extracting data from the seismic data (in SAC format) into a MATLAB matrix for
% easy processing of the data. The matrix is necessary to avoid extracting data from the
% SAC files for every data window interval. It is time efficient!

ii=0; % initializing parameter

for i=1:length(Stat_Net_Sta); % number of TA stations used in the research

Stat=Stat_Net_Sta(i,1); Net=Stat_Net_Sta(i,2); Sta=Stat_Net_Sta(i,3);
date_start=datenum(Stat_Net_Sta(i,4));date_end=datenum(Stat_Net_Sta(i,5));
LAT=lat_long(i,1); LONG=lat_long(i,2);

dt1=startTime0-date_start; dt2=date_end-startTime0;
% dt1 is the time between present date and start date of the TA dt2 is between the present
% date and the end date of the TA station. Both must be positive to be sure the present
% date is within the life-span of the TA station

if (dt1>0) && (dt2>0); % the present day is within the lifespan of the TA station

%       concatenation of the SAC filenames
Stat1 ='BHZ.M'; Net1 = char(Net); Sta1 = char(Sta);

 datafilename = sprintf('%s.%s.%s..%s%s',Stat11,Net1,Sta1,Stat1,ext2);
 %       e.g. 2012.319.00.00.00. 0000.TA.351A..BHZ.M.SAC
datafilename2 = sprintf('%s.%s.%s..%s%s',Stat12,Net1,Sta1,Stat1,ext2);
datafilename3 = sprintf('%s.%s.%s..%s%s',Stat13,Net1,Sta1,Stat1,ext2);

if exist(num2str(datafilename),'file')==0; % checking if the seismic data "datafilename" is
%        present in the folder. Returns 1 if present and 0 if not present.

continue % continue to the next filename if ‘datafilename’ is not present

else
    ii=ii+1;
    ReadSacBinary % function that reads the ‘datafilename’ into Matlab (waveform)
    [wr,wc]= size(waveform); % size of the waveform
    databank(1:wr,ii)=waveform; % the waveform of each TA stations are stored in a
% column number corresponding to their number on the TA stations list

% databank is a huge matrix containing waveforms of all the TA stations. The code does
% not need to extract data from the SAC format anymore but work with the databank

 end
```

```
 if exist(num2str(datafilename2),'file')==0; %% same as above. This is done the second
time because some of the datafilenames are broken into two or three data files with
different filenames. The files will be joined on the same column.

continue

 else

     datafilename=datafilename2;
     ReadSacBinary
     [wr2,wc2]= size(waveform);
     databank((wr+1):(wr+wr2),ii)=waveform; % join the second data to the first data.

 end

 if exist(num2str(datafilename3),'file')==0; % The third time in cases of three files

continue

 else

     datafilename=datafilename3;
     ReadSacBinary
     [wr3,wc3]= size(waveform);
     databank((wr+wr2+1):(wr+wr2+wr3),ii)=waveform;

 end
        end
   end


%%     Seismic processing sequence starts
jj=0; [dim1, dim]=size(databank);

for p=1:m; % iteration from one to the maximum number of data window
   ii=0; sum_noi=0;
    jj=jj+1;
distances=sprintf('%s_%s','dist',z);

fid9 = fopen(distances,'w'); % create a distance data file. This file will help sort the TA
% stations with their epicentral distances (Reference location: 30.1N -88.1E)
fclose(fid9);
startTime=startTime0+(2/24/60)*jj;
% 4-minute data window with 2 minute overlap to detect earthquakes at the boundaries.
endTime=endTime0+(2/24/60)*jj;
```

```
fid = fopen(filename1,'w');
fclose(fid);

for i=1:length(Stat_Net_Sta);
        ref_trc_p=zeros(9600,6); % creating a zero vector for the P-wave
        ref_trc_s=zeros(9600,6); % creating a zero vector for the S-wave
        Stat=Stat_Net_Sta(i,1); Net=Stat_Net_Sta(i,2); Sta=Stat_Net_Sta(i,3);
        date_start=datenum(Stat_Net_Sta(i,4));date_end=datenum(Stat_Net_Sta(i,5));
        LAT=lat_long(i,1); LONG=lat_long(i,2);

        dt1=startTime0-date_start;   dt2=date_end-startTime0;
        if (dt1>0) & (dt2>0);

              Stat1 ='BHZ.M';
              Net1 = char(Net);
              Sta1 = char(Sta);
              datafilename = sprintf('%s.%s.%s..%s%s',Stat11,Net1,Sta1,Stat1,ext2);

 if exist(num2str(datafilename),'file')==0;

continue

 else
          loc2=[LAT LONG]; % position of the station
          [km, nmi, mi] = haversine (loc1, loc2); % haversine is a function to determine the
%  distance between two points on the earth surface

          myformat = '%7.2f %7.2f\n';
          fid9 = fopen(distances,'a'); % open the file with permission to append
                     % write values at end of file
         fprintf(fid9, myformat, [ii km]);
          fclose(fid9);     % close the file
  end
        end
    end

%%      sorting the data with respects to their distances from the reference epicenter
   distance=load ([distances]);
   [Y,I]=sort(distance);
   dist=distance(:,2);
   dist1=sort(dist);

    u=round(p-1);
```

```
%% ----------------------------------------------------------------------------

   for i=1:dim; % dim is the number of TA station used

        Stat=Stat_Net_Sta(i,1); Net=Stat_Net_Sta(i,2); Sta=Stat_Net_Sta(i,3);
       date_start=datenum (Stat_Net_Sta(i,4));date_end=datenum(Stat_Net_Sta(i,5));
       LAT=lat_long (i,1); LONG=lat_long(i,2);

       dt1=startTime0-date_start;   dt2=date_end-startTime0;

       if (dt1>0) & (dt2>0);

            ii=ii+1; u=round(p-1);
            mytrace.data=databank((9601+4800*u):(19200+4800*u),ii);
            sampletimes=linspace(datenum(startTime),datenum(endTime),9600);
% determining the date number for the data window from startTime to endTime.

                if quest==0;

mytrace_noise.data=databank((1+4800*u):(9600+4800*u),ii);
% background noise using the previous 4 minute data

                else

mytrace_noise.data=databank((1+4800*(u-1)):(9600+4800*(u-1)),ii);
% when an earthquake has just been detected. Use the data in the 4-8 minute

                end

            loc2=[LAT LONG];% position of the TA station

%       filtering the dataset
[b,a]= butter(2,[(freq1/20),(freq2/20)]);
% Butterworth filter between freq1 and freq2 using 2 poles!!!
FILTERED=filtfilt(b,a,mytrace.data);  % applying the filter on the extracted signal.
% filtfilt applies the Butterworth filter twice in a way to make it a zero-phase filter.

FILTERED_noise=filtfilt(b,a,mytrace_noise.data);
% applying the filter on the extracted noise i.e. previous 4 minute data window

% determining the signal-to-noise ratio [snr=10*log (energy/energy_noise) in decibels]
peak_ampl=max(abs(FILTERED));% peak amplitude in the filtered signal.
energy_noise=mean2 (FILTERED_noise.^2);  % the mean of the squares (noise)
energy=mean2(FILTERED.^2);  % mean of the squares of each elements in the signal

snr=10*log(energy/energy_noise); % signal-to-noise ratio in decibel
```

```
% determining the distance between each TA station and the reference point
[km, nmi, mi] = haversine (loc1, loc2);

% determining the P and S wave moveout corrections. The equations were empirically
% determined using the traveltime to each TA station against epicentral distance for the
% November 10, 2012 and February 13, 2011 earthquakes. See thesis for details

% Empirical relationship between traveltimes and epicentral distance in the NGM
timeshift_pwave = (0.1449*km); % P-wave
timeshift_swave = (0.2587*km); % S-wave

km
Sta % just for monitoring the progress of the algorithm.

shift_p=ceil(40*(total_time-timeshift_pwave)); % determines the number of samples the
% waveform will be shifted to flattening the P wave before stacking
shifteddata_p0=circshift(FILTERED',[0 shift_p]);% shifts the waveform by shifted_p.

shift_s=ceil(40*(total_time-timeshift_swave)); % same as P- wave but now S- wave
shifteddata_s0=circshift(FILTERED',[0 shift_s]); % same as P- wave

x=tot-shift_p; % number of samples to get rid of after correcting for P-wave moveout

shifteddata_p0(x:tot)=0;% converts the shifted_p to zero
shifteddata_p=shifteddata_p0'; % convert to row vector

shifteddata_p_snr=shifteddata_p0(1:x-1); % extracting just the P- wave in the signal
energy_p=mean2(shifteddata_p_snr.^2); % P-wave energy
 snr_p=10*log(energy_p/energy_noise); % P-wave signal-to-noise ratio

xs=tot-shift_s; % same as P wave
shifteddata_s0(xs:tot)=0; % same as P wave
shifteddata_s=shifteddata_s0'; % same as P wave

shifteddata_s_snr=shifteddata_s0(1:xs-1); %SAME AS P WAVE

energy_s=mean2(shifteddata_s_snr.^2);% S-wave energy
snr_s=10*log(energy_s/energy_noise);% S-wave signal-to-noise ratio

% Determining the envelope function using Hilbert Transform of the waveforms. This
% will avoid destructive stacking of the waveform data
analy_n=hilbert(FILTERED_noise); % this is a complex number
y_noise0=abs(analy_n); % amplitude spectrum of the Hilbert Transform
y_noise1=(y_noise0.^2).^(1/2); % Background noise envelope
noise_max=max(y_noise1);
```

```
analy_p=hilbert(shifteddata_p);% Hilbert transform of the P wave for the envelope
function
y_p0=abs(analy_p);% absolute of the envelope function
y_p1=(y_p0.^2).^(1/2); % P-wave envelope
y_p_max=max(y_p1);

analy_s=hilbert(shifteddata_s);% Hilbert transform of the S wave
y_s0=abs(analy_s);% absolute of the envelope function
y_s1=(y_s0.^2).^(1/2); % S-wave envelope
y_s_max=max(y_s1);

%       dimension of resulting envelope waveforms
[dim_sam1, dim_sam2]=size(sampletimes); % sample times for display
[dim_noi, dim1_c]=size(y_noise1); % background noise
[dim_p, dim1_c]=size(y_p1); % P- wave
[dim_s, dim1_c]=size(y_s1); % S-wave

myformat = '%7.2f %7.2f %7.2f %7.2f %7.2f\n\n'; % format the fprintf will print to file

%       open the file with permission to append
fid = fopen(filename1,'a'); % write values at end of file
fprintf(fid, myformat, [y_p_max y_s_max noise_max snr_p snr_s]);
% print using myformat
fclose(fid); % close the file
if i==1; % all vector should be zero initially
   y_noise2=zeros(size(y_noise1));
   y_p2=zeros(size(y_p1));
   y_s2=zeros(size(y_s1));
end

% applying weighted stacking method using the inverse-square of the standard deviation
% as the weight

y_noise2=y_noise2+y_noise1.*(std(y_noise1))^(-2);
y_p2=y_p2+y_p1.*(std(y_noise1))^(-2);
y_s2=y_s2+y_s1.*(std(y_noise1))^(-2);

else
% i.e. when (dt1 and dt2 are less than zero. i.e. the time to be processed is not within the
% lifespan of the TA station to avoid crashing of the program. This enables the program
% to work with any TA data without input from the user during processing. It is purely
% automated after the start of the program.

continue % continue to the next TA station without stopping the program.
end
```

```
end % finish with all the TA stations

current_time=[datestr(startTime)  datestr(endTime)]
% display the current time for the data window to monitor the progress of the program

std_data=std(load ([filename1]));

std_p=std_data(1,1); %standard deviation of y_p_max
std_s=std_data(1,2); %standard deviation of y_s_max
std_noi=std_data(1,3); %standard deviation of noise_max
std_snr_p=std_data(1,4); %standard deviation of snr_p
std_snr_s=std_data(1,5); %standard deviation of snr_s

% peak amplitude of P- and S- waves, and background noise
peak_ampl_noise=max(y_noise2);
peak_ampl_p=max(y_p2);
peak_ampl_s=max(y_s2);

energy_noise2=mean2(y_noise2.^2);
energy_p2=max(y_p2.^2); % P-wave
energy_s2=max(y_s2.^2); % S-wave energy

snr_p2=10*log10(energy_p2/energy_noise2); % signal-to-noise ratio
snr_s2=10*log10(energy_s2/energy_noise2);
y_noise3=y_noise2'; % matrix transform
y_p3=y_p2';
y_s3=y_s2';

% determining the P- and S- wave lamdas. Lamda is the ratio of the standard deviation of
% the signal to the standard deviation of the background noise. This index will be used to
% detect earthquakes in conjunction with the signal-to-noise ratio
lamda_p= std_p/std_noi; %P-wave lamda
lamda_s=std_s/std_noi;  %S-wave lamda
myformat2 = '%7.2f %7.2f %7.2f %7.2f %7.2f\n\n';
fid2 = fopen(filename2,'a'); % open the file with permission to append
% write values at end of file
fprintf(fid2, myformat2, [snr_p snr_s std_p std_s std_noi]);
% store the signal-to-noise ratios and the standard deviations in a file
fclose(fid2); % close the file

myformat4 = '%s %s\n\n';
fid4 = fopen(filename4,'a'); % open the file with permission to append
% write values at end of file
fprintf(fid4, myformat4, [datestr(startTime)  datestr(endTime)]);
% store the start and end time of the processed data window
```

```
fclose(fid4); % close the file

if (snr_p2>10) || (snr_s2>10) %(lamda_p>2.5) & (lamda_s>2.5) %; declare an event
% when the signal-to-noise ratio is greater than 10
quest=1; % store 1 in quest if an event is just declared
            myformat3 = '%s %s\n';
            fid3 = fopen(filename3,'a');
 % write values at end of file
fprintf(fid3, myformat3, [datestr(startTime+(2/24/60))  datestr(endTime+(2/24/60))]);
% store the start and end time where the event was declared
fclose(fid3); % close the file

h1 = figure;
subplot(3,1,1)
plot(sampletimes,y_noise3,'b');
title(['Noise Waveforms (STD of Peak Amp=  ',num2str(std_noi),' ) Date
StartTime:',datestr(startTime)]);
grid on; datetick;

subplot(3,1,2)
plot(sampletimes,y_p3,'b');
title(['      P-wave Signal (SNR=  ',num2str(snr_p2),' dB; std of Peak Amp=
',num2str(std_p),' ;  Epicenter =',num2str(lat1),'N, ',num2str(long1),'E )']);
grid on; datetick;

subplot(3,1,3)
plot(sampletimes,y_s3,'b');
title(['      S-wave Signal (SNR=  ',num2str(snr_s2),' dB;  std of Peak Amp=
',num2str(std_s),'  ; Epicenter =',num2str(lat1),'N, ',num2str(long1),'E )']);
grid on; datetick;

print(h1, '-dpsc','-noui','-append',pdf_name); % save the figures in a postscript file
beep

%% displaying the data window where the earthquake was detected at all the TA stations
t_n=[0:(4/9600):4]; % time for the data window (9600 samples = 4 minute data)
t_n=t_n(2:9601);
Data= databank((9601+4800*u):(19200+4800*u),:); % data for the earthquake

IMAGE=filtfilt(b,a,Data); % filtering the data

tt=max(IMAGE); % maximum value in the data window.
ttt=tt(:,I(:,2)); % sorting the TA waveforms by distance to the Gulf.
% I is a matrix containing the rank of the epicentral distances
FILTERED1=IMAGE(:,I(:,2)); % save the filtered and sorted waveforms in FILTERED1
```

```
scale=1; % no exaggeration!

% applying the RMS AGC gain function
desired_rms=10000; % constant
l_t=length(dist1);
l=dist1(l_t);

k_t=[0:50:450];

for i=0:50:450
   g_t(i/50+1)=desired_rms/rms(ttt(find(dist1>=i & dist1<=(i+50))));
% determining the gain in 50 km interval to create the gain function in the data window
end

g_t=g_t(2:10);
g = [g_t(1) g_t max(g_t)]; % full gain function including the last part of the interval
k=[k_t l]; % epicentral axis
p= interp1(k,g,dist1); % interpolation of the gain value at each distance interval
ll=(ttt+fliplr(ttt))/2;

for r=1:dim
   AGC(:,r)=p(r)*FILTERED1(:,r); % multiply the gain with each TA waveform
end

%       displaying the gained waveforms
h2=figure ;
section_display(AGC,scale,dist1',t_n) % function for data presentation
ylabel('time (min) ');
xlabel(['[ ', datestr(startTime) ,' to ',datestr(endTime) ,' ]  Index = ', num2str(u)]);
title([' Seismic Waveforms (AGC) with Epicentral Distance ( f =  ',num2str(freq1),' - ',
num2str(freq2),' Hz )']);

%       store the events detected within certain numbers to be recorded in postscript files
if u>0 && u<=500;

print(h2, '-dpsc','-noui','-append',pdf_name21);
% store the detected events between interval 0 and 500 into filename pdf_name21

elseif u>500 && u<=1000; print(h2, '-dpsc','-noui','-append',pdf_name22);
% Index between 500 and 1000
elseif u>1000 && u<=1500; print(h2, '-dpsc','-noui','-append',pdf_name23);
elseif u>1500 && u<=2000; print(h2, '-dpsc','-noui','-append',pdf_name24);
elseif u>2000 && u<=2500; print(h2, '-dpsc','-noui','-append',pdf_name25);
elseif u>2500 && u<=3000; print(h2, '-dpsc','-noui','-append',pdf_name26);
elseif u>3000 && u<=3500; print(h2, '-dpsc','-noui','-append',pdf_name27);
elseif u>3500 && u<=4000; print(h2, '-dpsc','-noui','-append',pdf_name28);
```

```
elseif u>4000 && u<=4500; print(h2, '-dpsc','-noui','-append',pdf_name29);
elseif u>4500 && u<=5000; print(h2, '-dpsc','-noui','-append',pdf_name210);
elseif u>5000 && u<=5500; print(h2, '-dpsc','-noui','-append',pdf_name211);
else
print(h2, '-dpsc','-noui','-append',pdf_name212);
end
    else
        quest=0; % if there is no earthquake, return the value of zero (0) in quest
        beep off
    end
i=1;
end
toc % dislay the time required to finish the process!
```

```
function [km nmi mi] = haversine(loc1, loc2)
% HAVERSINE     Compute distance between locations using Haversine formula
%   KM = HAVERSINE(LOC1, LOC2) returns the distance KM in km between
%   locations LOC1 and LOC2 using the Haversine formula.  LOC1 and LOC2 are
%   latitude and longitude coordinates that can be expressed as either
%   strings representing degrees, minutes, and seconds (suffixed with
%   N/S/E/W), or numeric arrays representing decimal degrees (where
%   negative indicates West/South).
%
%   [KM, NMI, MI] = HAVERSINE(LOC1, LOC2) returns the computed distance in
%   kilometers (KM), nautical miles (NMI), and miles (MI).
%
%   Examples
%       haversine('53 08 50N, 001 50 58W', '52 12 16N, 000 08 26E') returns
%           170.2547
%       haversine([53.1472 -1.8494], '52 12.16N, 000 08.26E') returns
%           170.2508
%       haversine([53.1472 -1.8494], [52.2044 0.1406]) returns 170.2563
%
%   Inputs
%       LOC must be either a string specifying the location in degrees,
%       minutes and seconds, or a 2-valued numeric array specifying the
%       location in decimal degrees.  If providing a string, the latitude
%       and longitude must be separated by a comma.
%
%   Notes
%       The Haversine formula is used to calculate the great-circle
%       distance between two points, which is the shortest distance over
%       the earth's surface.
%
```

```
%       This program was created using equations found on the website
%       http://www.movable-type.co.uk/scripts/latlong.html

% Created by Josiah Renfree
% May 27, 2010

%% Check user inputs

% If two inputs are given, display error
if ~isequal(nargin, 2)
    error('User must supply two location inputs')

% If two inputs are given, handle data
else

    locs = {loc1 loc2};     % Combine inputs to make checking easier
    % Cycle through to check both inputs
    for i = 1:length(locs)

        % Check inputs and convert to decimal if needed
        if ischar(locs{i})

            % Parse lat and long info from current input
            temp = regexp(locs{i}, ',', 'split');
            lat = temp{1}; lon = temp{2};
            clear temp
            locs{i} = [];           % Remove string to make room for array

            % Obtain degrees, minutes, seconds, and hemisphere
            temp = regexp(lat, '(\d+)\D+(\d+)\D+(\d+)(\w?)', 'tokens');
            temp = temp{1};

            % Calculate latitude in decimal degrees
            locs{i}(1) = str2double(temp{1}) + str2double(temp{2})/60 + ...
                str2double(temp{3})/3600;

            % Make sure hemisphere was given
            if isempty(temp{4})
                error('No hemisphere given')

            % If latitude is south, make decimal negative
            elseif strcmpi(temp{4}, 'S')
                locs{i}(1) = -locs{i}(1);
            end

            clear temp
```

```
            % Obtain degrees, minutes, seconds, and hemisphere
            temp = regexp(lon, '(\d+)\D+(\d+)\D+(\d+)(\w?)', 'tokens');
            temp = temp{1};

            % Calculate longitude in decimal degrees
            locs{i}(2) = str2double(temp{1}) + str2double(temp{2})/60 + ...
                str2double(temp{3})/3600;

            % Make sure hemisphere was given
            if isempty(temp{4})
                error('No hemisphere given')

            % If longitude is west, make decimal negative
            elseif strcmpi(temp{4}, 'W')
                locs{i}(2) = -locs{i}(2);
            end

            clear temp lat lon
        end
    end
end

% Check that both cells are a 2-valued array
if any(cellfun(@(x) ~isequal(length(x),2), locs))
    error('Incorrect number of input coordinates')
end

% Convert all decimal degrees to radians
locs = cellfun(@deg2rad, locs, 'UniformOutput', 0);

%% Begin calculation

R = 6371;                                  % Earth's radius in km
delta_lat = locs{2}(1) - locs{1}(1);       % difference in latitude
delta_lon = locs{2}(2) - locs{1}(2);       % difference in longitude
a = sin(delta_lat/2)^2 + cos(locs{1}(1)) * cos(locs{2}(1)) * ...
    sin(delta_lon/2)^2;
c = 2 * atan2(sqrt(a), sqrt(1-a));
km = R * c;                                % distance in km

%% Convert result to nautical miles and miles

nmi = km * 0.539956803;                    % nautical miles
mi = km * 0.621371192;                     % miles
```

```
function [lat_long, Stat_Net_Sta] = importfile(fileToRead1)
%IMPORTFILE(FILETOREAD1)
%  Imports data from the specified file
%  FILETOREAD1:  file to read
%  Auto-generated by MATLAB on 22-Jan-2014 20:39:04

% Import the file
newData1 = importdata(fileToRead1);
lat_long=newData1.data(:,:);
Stat_Net_Sta=newData1.textdata(:,:);


function ReadSacBinary(datafilename)
%
% Matlab m-file to read Binary SAC files
%
% This file is currently set up to read only evenly spaced waveform files.
%
% First, get the name of the file
% filnam=input('Give the name of the SAC binary file: ','s');
 filnam=datafilename;

%  Open the file
          [fid,message]=fopen(filnam,'r','b');
%           [fid,message]=fopen(filnam,'r','l'); % The 'b' argument is for a big-endian file
% Note:  Because SAC files were designed for SUN computers, which are
%  "big-endian" machines, all SAC files are written with the binary data
%  in big-endian format.  On a PC, the natural way that numbers are stored
%  is in little-endian format.  Thus, when a SAC file is opened on a PC,
%  the "open" statement must include a qualifier indicating that the binary
%  data file was written in big-endian format.  If this macro is being run
%  on a big-endian machine (like a Sun, Mac, HP, IBM RS or SGI), then the
%  'b' argument in the open statement is optional.
%
if fid == -1  % Oops, there was an error and no file was opened
    info='Cannot open file:'
    filnam
    message
  else
    % We successfully opened the file, so read it

    % Parse out all the SAC header variables
    [delta,count]=fread(fid,1,'float32');
    [depmin,count]=fread(fid,1,'float32');
    [depmax,count]=fread(fid,1,'float32');
    [scale,count]=fread(fid,1,'float32');
```

```
[odelta,count]=fread(fid,1,'float32');
[b,count]=fread(fid,1,'float32');
[e,count]=fread(fid,1,'float32');
[o,count]=fread(fid,1,'float32');
[a,count]=fread(fid,1,'float32');
[junk1,count]=fread(fid,1,'float32');
[t0,count]=fread(fid,1,'float32');
[t1,count]=fread(fid,1,'float32');
[t2,count]=fread(fid,1,'float32');
[t3,count]=fread(fid,1,'float32');
[t4,count]=fread(fid,1,'float32');
[t5,count]=fread(fid,1,'float32');
[t6,count]=fread(fid,1,'float32');
[t7,count]=fread(fid,1,'float32');
[t8,count]=fread(fid,1,'float32');
[t9,count]=fread(fid,1,'float32');
[f,count]=fread(fid,1,'float32');
[resp0,count]=fread(fid,1,'float32');
[resp1,count]=fread(fid,1,'float32');
[resp2,count]=fread(fid,1,'float32');
[resp3,count]=fread(fid,1,'float32');
[resp4,count]=fread(fid,1,'float32');
[resp5,count]=fread(fid,1,'float32');
[resp6,count]=fread(fid,1,'float32');
[resp7,count]=fread(fid,1,'float32');
[resp8,count]=fread(fid,1,'float32');
[resp9,count]=fread(fid,1,'float32');
[stla,count]=fread(fid,1,'float32');
[stlo,count]=fread(fid,1,'float32');
[stel,count]=fread(fid,1,'float32');
[stdp,count]=fread(fid,1,'float32');
[evla,count]=fread(fid,1,'float32');
[evlo,count]=fread(fid,1,'float32');
[evel,count]=fread(fid,1,'float32');
[evdp,count]=fread(fid,1,'float32');
[junk2,count]=fread(fid,1,'float32');
[user0,count]=fread(fid,1,'float32');
[user1,count]=fread(fid,1,'float32');
[user2,count]=fread(fid,1,'float32');
[user3,count]=fread(fid,1,'float32');
[user4,count]=fread(fid,1,'float32');
[user5,count]=fread(fid,1,'float32');
[user6,count]=fread(fid,1,'float32');
[user7,count]=fread(fid,1,'float32');
[user8,count]=fread(fid,1,'float32');
[user9,count]=fread(fid,1,'float32');
```

```
[dist,count]=fread(fid,1,'float32');
[az,count]=fread(fid,1,'float32');
[baz,count]=fread(fid,1,'float32');
[gcarc,count]=fread(fid,1,'float32');
[junk3,count]=fread(fid,1,'float32');
[junk4,count]=fread(fid,1,'float32');
[depmen,count]=fread(fid,1,'float32');
[cmpaz,count]=fread(fid,1,'float32');
[cmpinc,count]=fread(fid,1,'float32');
[junk5,count]=fread(fid,1,'float32');
[junk6,count]=fread(fid,1,'float32');
[junk7,count]=fread(fid,1,'float32');
[junk8,count]=fread(fid,1,'float32');
[junk9,count]=fread(fid,1,'float32');
[junk10,count]=fread(fid,1,'float32');
[junk11,count]=fread(fid,1,'float32');
[junk12,count]=fread(fid,1,'float32');
[junk13,count]=fread(fid,1,'float32');
[junk14,count]=fread(fid,1,'float32');
[junk15,count]=fread(fid,1,'float32');
[nzyear,count]=fread(fid,1,'uint32');
[nzjday,count]=fread(fid,1,'uint32');
[nzhour,count]=fread(fid,1,'uint32');
[nzmin,count]=fread(fid,1,'uint32');
[nzsec,count]=fread(fid,1,'uint32');
[nzmsec,count]=fread(fid,1,'uint32');
[nvhdr,count]=fread(fid,1,'uint32');
[junk16,count]=fread(fid,1,'uint32');
[junk17,count]=fread(fid,1,'uint32');
[npts,count]=fread(fid,1,'uint32');
[junk17,count]=fread(fid,1,'uint32');
[junk18,count]=fread(fid,1,'uint32');
[junk19,count]=fread(fid,1,'uint32');
[junk20,count]=fread(fid,1,'uint32');
[junk21,count]=fread(fid,1,'uint32');
[iftype,count]=fread(fid,1,'uint32');
[idep,count]=fread(fid,1,'uint32');
[iztype,count]=fread(fid,1,'uint32');
[junk22,count]=fread(fid,1,'uint32');
[iinst,count]=fread(fid,1,'uint32');
[istreg,count]=fread(fid,1,'uint32');
[ievreg,count]=fread(fid,1,'uint32');
[ievtyp,count]=fread(fid,1,'uint32');
[iqual,count]=fread(fid,1,'uint32');
[isynth,count]=fread(fid,1,'uint32');
[junk23,count]=fread(fid,1,'uint32');
```

```
[junk24,count]=fread(fid,1,'uint32');
[junk25,count]=fread(fid,1,'uint32');
[junk26,count]=fread(fid,1,'uint32');
[junk27,count]=fread(fid,1,'uint32');
[junk28,count]=fread(fid,1,'uint32');
[junk29,count]=fread(fid,1,'uint32');
[junk30,count]=fread(fid,1,'uint32');
[junk31,count]=fread(fid,1,'uint32');
[junk32,count]=fread(fid,1,'uint32');
[leven,count]=fread(fid,1,'uint32');
[lspol,count]=fread(fid,1,'uint32');
[lovrok,count]=fread(fid,1,'uint32');
[lcalda,count]=fread(fid,1,'uint32');
[junk33,count]=fread(fid,1,'uint32');
[ktemp,count]=fread(fid,8,'schar');
kstnm=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kevnm=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kjunk=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
khole=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
ko=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
ka=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt0=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt1=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt2=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt3=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt4=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt5=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt6=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt7=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt8=char(ktemp');
[ktemp,count]=fread(fid,8,'schar');
kt9=char(ktemp');
```

```
    [ktemp,count]=fread(fid,8,'schar');
    kf=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kuser0=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kuser1=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kuser2=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kcmpnm=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    knetwk=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kdatrd=char(ktemp');
    [ktemp,count]=fread(fid,8,'schar');
    kinst=char(ktemp');
    % Done reading the header, now read the data
    [waveform,countw]=fread(fid,inf,'float32');
end  %  End the if loop
fclose(fid);  % Close the file
%
% Send some basic info to the Matlab Command window
% info = ['Station = ',kstnm,',  Comp = ',kcmpnm]
% info = ['Number of points = ',num2str(npts),',  DT = ',num2str(delta)]
% info = 'The seismogram is in the array: "waveform"'


function section_display (a,scal,x,z)
% mwigb: is a modified version of wigb.m which plot seismic data using
% wiggles but displays the screen as a normal seismic display such as those
% seen using SeismicUnix.
%
%  wigb(a,scal,x,z,amx)
%
%  IN    a: seismic data
%        scale: multiple data by scale
%        x: x-axis;
%        z: vertical axis (time or depth)
%    x and z are vectors with offset and time.
%
%    If only 'a' is enter, 'scal,x,z,amn,amx' are decided automatically;
%    otherwise, 'scal' is a scalar; 'x, z' are vectors for annotation in
%    offset and time, amx are the amplitude range.
%
% Author:
%   Xingong Li, Dec. 1995
% Changes:
```

```
%   Jun11,1997: add amx
%   May16,1997: updated for v5 - add 'zeros line' to background color
%   May17,1996: if scal ==0, plot without scaling
%   Aug6, 1996: if max(tr)==0, plot a line
%   Aug12, 2011: modification by Wail A. Mousa and Abdullatif A. Al-Shuhail
%   July 31, 2015: modified by Oluwaseun Fadugba for earthquake data display

if nargin == 0, nx=10;nz=10; a = rand(nz,nx)-0.5; end;

[nz,nx]=size(a);

trmx= max(abs(a));
amx=mean(trmx);
if (nargin <= 2); x=[1:nx]; z=[1:nz]; end;
if (nargin <= 1); scal =1; end;

 if nx <= 1; disp(' ERR:PlotWig: nx has to be more than 1');return;end;

 % take the average as dx
    dx1 = abs(x(2:nx)-x(1:nx-1));

    dx = median(dx1);

 dz=z(2)-z(1);
 xmx=max(max(a)); xmn=min(min(a));

 if scal == 0; scal=1; end;
 a = a * dx /amx;
 a = a * scal;

 fprintf(' PlotWig: data range [%f, %f], plotted max %f \n',xmn,xmx,amx);

% set display range
x1=min(x)-2.0*dx; x2=max(x)+2.0*dx;
z1=min(z)-dz; z2=max(z)+dz;

set(gca,'NextPlot','add','Box','on', ...
  'XLim', [x1 x2], ...
  'YDir','reverse', ...
  'YLim',[z1 z2]);

     fillcolor = [1 1 1];
    linecolor = [0 0 0];
    linewidth = 0.2;

    z=z';   % input as row vector
```

```
   zstart=z(1);
   zend  =z(nz);

for i=1:nx,

  if trmx(i) ~= 0;    % skip the zero traces
   tr=a(:,i);  % --- one scale for all section

pp=polyfit(z,tr,0); % fitting an horizontal line to pass through the data set to correct for
the shift from the zero line.

   tr=tr-pp; % shifting the dataset back to the zero line. Now we have positive and
negative amplitudes

   s = sign(tr) ;
   i1= find( s(1:nz-1) ~= s(2:nz) );   % zero crossing points
   npos = length(i1);


   %12/7/97
   zadd = i1 + tr(i1) ./ (tr(i1) - tr(i1+1)); %locations with 0 amplitudes
   aadd = zeros(size(zadd));

   [zpos,vpos] = find(tr >0);
   [zz,iz] = sort([zpos; zadd]);   % indices of zero point plus positives
   aa = [tr(zpos); aadd];
   aa = aa(iz);

   % be careful at the ends

   pp=polyfit(z,tr,0); % fitting an horizontal line to pass through the data set to correct for
the shift from the zero line.

   tr=tr-pp; % shifting the dataset back to the zero line. Now we have positive and
negative amplitudes

      if tr(1)>0,     a0=0; z0=1.00;
      else
         if numel(zadd)>0;
         a0=0; z0=zadd(1);
         else
            continue
         end
      end;
      if tr(nz)>0,    a1=0; z1=nz;
      else
```

```
          if numel(zadd)>0;
          a1=0; z1=max(zadd);
          else

             continue
          end
       end;

    zz = [z0; zz; z1; z0];
    aa = [a0; aa; a1; a0];

    zzz = zstart + zz*dz -dz;

    patch( aa+x(i) , zzz,  fillcolor);

    line( 'Color',[1 1 1],'EraseMode','background',  ...
        'Xdata', x(i)+[0 0], 'Ydata',[zstart zend]); % remove zero line

%'LineWidth',linewidth, ...
%12/7/97 'Xdata', x(i)+[0 0], 'Ydata',[z0 z1]*dz);  % remove zero line

    line( 'Color',linecolor,'EraseMode','background',  ...
     'LineWidth',linewidth, ...
     'Xdata', tr+x(i), 'Ydata',z);  % negatives line

   else % zeros trace
    line( 'Color',linecolor,'EraseMode','background',  ...
     'LineWidth',linewidth, ...
        'Xdata', [x(i) x(i)], 'Ydata',[zstart zend]);
   end;
end;

% Displaying the seismic gather in the traditional way for displaying seismic data
% Aug 12, 2011
x=10;y=10;%starting screen position
w=600;%figure width
h=1200;%figure hieght
% set(gcf,'position',[x y w h]);
% Putting the spatial axis in the top of the figure
set(gca,'xaxislocation','top')
```

Appendix C

Algorithm for plotting data points on Google Map

```
%%     Data
DATA1=xlsread('NOV10_Jacknife_model1.xls');
 % files containing the locations determined from the jackknife analysis
DATA2=xlsread('NOV10_Jacknife_model2.xls');
DATA3=xlsread('NOV10_Jacknife_model3.xls');
DATA4=xlsread('NOV10_Jacknife_model4.xls');
DATA5=xlsread('NOV10_Jacknife_model5.xls');
DATA6=xlsread('NOV10_Jacknife_model6.xls');
%%
GAS_LOC=xlsread('Gas_well_loc_AL.xlsx'); % location of the drilled wells
% plotting the different drilled wells according to their statuses e.g. producing, plug in etc
latG=GAS_LOC(1:7,2); lonG=GAS_LOC(1:7,1); plot(lonG,latG,'.c','MarkerSize',20);
hold on
latG=GAS_LOC(8:20,2); lonG=GAS_LOC(8:20,1); plot(lonG,latG,'.k','MarkerSize',20);
hold on
latG=GAS_LOC(21:42,2); lonG=GAS_LOC(21:42,1);
plot(lonG,latG,'.b','MarkerSize',20); hold on
latG=GAS_LOC(43:45,2); lonG=GAS_LOC(43:45,1);
plot(lonG,latG,'.y','MarkerSize',20); hold on
latG=GAS_LOC(46:49,2); lonG=GAS_LOC(46:49,1);
plot(lonG,latG,'.m','MarkerSize',20); hold on

%% parameters of the mean epicenter and the standard deviation of the 68% uncertainty
meanr=12; meanc=13; meanc2=14;
lat2=30.15316667;   lon2=-88.08216667; sd2= 5.000031805;
lat1=30.1935;   lon1=-88.0345; sd1=5.329639269;
lat3=30.32366667;   lon3=-87.99283333; sd3=2.668714669;
lat6=30.19216667;   lon6=-88.05933333; sd6= 7.952435791;
lat5=30.1905;   lon5=-88.0595; sd5=7.024533065;

%%     Ploting
lat1B=DATA1(meanr,meanc); lon1B=DATA1(meanr,meanc2);
plot(lon1B,lat1B,'*g','MarkerSize',20); hold on
lat2B=DATA2(meanr,meanc); lon2B=DATA2(meanr,meanc2);
plot(lon2B,lat2B,'*r','MarkerSize',20); hold on
lat3B=DATA3(meanr,meanc); lon3B=DATA3(meanr,meanc2);
plot(lon3B,lat3B,'*k','MarkerSize',20); hold on
lat4B=DATA4(meanr,meanc); lon4B=DATA4(meanr,meanc2);
plot(lon4B,lat4B,'*c','MarkerSize',20); hold on
lat5B=DATA5(meanr,meanc); lon5B=DATA5(meanr,meanc2);
plot(lon5B,lat5B,'*y','MarkerSize',20); hold on
lat6B=DATA6(meanr,meanc); lon6B=DATA6(meanr,meanc2);
plot(lon6B,lat6B,'*m','MarkerSize',20); hold on
```

```
% plotting the 68% and 95% uncertainties in epicenter determination around the
% epicenters using different velocity models

[latc2,longc2]=scircle1(lat2,lon2,(sd2*2),[],earthRadius('km'));
hold on; plot(longc2,latc2,'r') % plotting the 68% uncertainty in epicenter determination
[latc2,longc2]=scircle1(lat2,lon2,sd2,[],earthRadius('km'));
hold on; plot(longc2,latc2,'b'); hold on % plotting the 95% uncertainty
latb2=30.11;lonb2=-88.10;plot(lonb2,latb2,'.b','MarkerSize',30) % the mean epicenter


[latc6,longc6]=scircle1(lat6,lon6,(sd6*2),[],earthRadius('km')); hold on;
plot(longc6,latc6,'r')
[latc6,longc6]=scircle1(lat6,lon6,sd6,[],earthRadius('km')); hold on; plot(longc6,latc6,'b');
hold on

[latc3,longc3]=scircle1(lat3,lon3,(sd3*2),[],earthRadius('km')); hold on;
plot(longc3,latc3,'r')
[latc3,longc3]=scircle1(lat3,lon3,sd3,[],earthRadius('km')); hold on; plot(longc3,latc3,'b');
hold on

[latc1,longc1]=scircle1(lat1,lon1,(sd1*2),[],earthRadius('km')); hold on;
plot(longc1,latc1,'r')
[latc1,longc1]=scircle1(lat1,lon1,sd1,[],earthRadius('km')); hold on; plot(longc1,latc1,'b');
hold on

[latc5,longc5]=scircle1(lat5,lon5,(sd5*2),[],earthRadius('km')); hold on;
plot(longc5,latc5,'r')
[latc5,longc5]=scircle1(lat5,lon5,sd5,[],earthRadius('km')); hold on; plot(longc5,latc5,'b');
hold on

lat1=DATA1(1:(meanc-1),meanc); lon1=DATA1(1:(meanc-1),meanc2);
plot(lon1,lat1,'*g','MarkerSize',10); hold on
lat2=DATA2(1:(meanc-1),meanc); lon2=DATA2(1:(meanc-1),meanc2);
plot(lon2,lat2,'*r','MarkerSize',10); hold on
lat3=DATA3(1:(meanc-1),meanc); lon3=DATA3(1:(meanc-1),meanc2);
plot(lon3,lat3,'*k','MarkerSize',10); hold on
lat4=DATA4(1:(meanc-1),meanc); lon4=DATA4(1:(meanc-1),meanc2);
plot(lon4,lat4,'*c','MarkerSize',10); hold on
lat5=DATA5(1:(meanc-1),meanc); lon5=DATA5(1:(meanc-1),meanc2);
plot(lon5,lat5,'*y','MarkerSize',10); hold on
lat6=DATA6(1:(meanc-1),meanc); lon6=DATA6(1:(meanc-1),meanc2);
plot(lon6,lat6,'*m','MarkerSize',10); hold on

lat1B=DATA1(meanr,meanc);
lon1B=DATA1(meanr,meanc2);plot(lon1B,lat1B,'*g','MarkerSize',30);hold on
```

```
lat2B=DATA2(meanr,meanc);
lon2B=DATA2(meanr,meanc2);plot(lon2B,lat2B,'*r','MarkerSize',30);hold on
lat3B=DATA3(meanr,meanc);
lon3B=DATA3(meanr,meanc2);plot(lon3B,lat3B,'*k','MarkerSize',30);hold on
lat4B=DATA4(meanr,meanc);
lon4B=DATA4(meanr,meanc2);plot(lon4B,lat4B,'*c','MarkerSize',30);hold on
lat5B=DATA5(meanr,meanc);
lon5B=DATA5(meanr,meanc2);plot(lon5B,lat5B,'*y','MarkerSize',30);hold on
lat6B=DATA6(meanr,meanc);
lon6B=DATA6(meanr,meanc2);plot(lon6B,lat6B,'*m','MarkerSize',30);hold on

%%     Boundaries
latb=30.5; lonb=-88;plot(lonb,latb,'.r','MarkerSize',1);hold on
latb2=30.11;lonb2=-88.10;plot(lonb2,latb2,'.b','MarkerSize',1)

makescale; plot_google_map

title('NOV 10 2012'); legend('Dry and Abandoned','Plugged and
Abandoned','Producing','Shut In','Temporarily Abandoned','MODEL 1','MODEL
2','MODEL 3','MODEL 4','MODEL 5','MODEL 6','95% Confidence','68%
Confidence','Previous Epicenter (USGS)'); %,'Nov 10 EQ (USGS)'


function h = makescale(varargin)
%MAKESCALE creates a scale for map data.
%
%   MAKESCALE creates a scale on the current axis based on the current axis
%      limits. The scale is made to occupy 1/5th of the map. It is placed
%      in the southeast corner of the map. The units will either be in
%      milimeters, meters or kilometers, depending on the size of the map.
%
%   MAKESCALE(H_AXIS) creates a scale on the axis specificed by the handle
%      H_AXIS based on the its axes limits. H_AXIS must be a scalar.
%
%   MAKESCALE(SCALE) creates a scale made to occupy 1/SCALE of the map.
%      SCALE must be a scalar, and is bounded to be between 1.1 and 10. If
%      a larger value is passed in, 10 will be used. If a smaller value is
%      passed in, 1.1 will be used.
%
%   MAKESCALE(LOCATION) places the scale in the location specified by
%      LOCATION. Acceptable values for location are as follows
%          'northeast'     'ne'
%          'northwest'     'nw'
%          'southeast'     'se'
%          'southwest'     'sw'
%          'north'         'n'
```

```
%          'south'         's'
%
%   MAKESCALE('units',UNITS) changes the units systems from SI to imperical
%      units. UNITS should be either 'si' or 'imp.' The units displayed
%      are automatically switched between milimeters, meters, and
%      kilometers for the SI system, or between inches, feet, and statuate
%      miles for the imperical system.
%
%   H = MAKESCALE(...) outputs H, a 3x1 containing the handles of the of
%      box, line, and text.
%
%   Any number of these input sets may be passed in any order.
%
%   The map scale will automatically be updated as the figure is zoomed,
%      panned, resized, or clicked on. It will not, however, be updated
%      upon using the commands "axis", "xlim", or "ylim" as these do not
%      have callback functionality.
%
%   Example:
%      load conus
%      figure
%      plot(uslon,uslat);
%      makeScale
%
%   Example: Placed in the south
%      load conus
%      figure
%      plot(uslon,uslat);
%      makeScale('south')
%
%   Example: Half the size of the Window
%      load conus
%      figure
%      plot(uslon,uslat);
%      makeScale(2,'south')
%
%   Example: Use Imperical Units
%      load conus
%      figure
%      plot(uslon,uslat);
%      makeScale(2,'south','units','imp')
%
%   Example: Zooming In
%      load conus
%      figure
%      plot(uslon,uslat);
```

```
%       makeScale(2,'south')
%       zoom(2)
%
%   Note: This assumes axis limits are in degrees. The scale is sized
%       correctly for the center latitude of the map. As the size of
%       degrees longitude change with latitude, the scale becomes invalid
%       with very large maps. Spherical Earth is assumed. Ideally, the map
%       will be proportioned correctly in order to reflect the relationship
%       between a degree latitude and a degree longitude at the center of
%       the map.
%
% By Jonathan Sullian - October 2011

% Check to make sure the correct number of inputs are passed in.
error(nargchk(0,5,nargin,'struct'));

% Parse Inputs
[anum,latlim,lonlim,scale,location,units] = parseInputs(varargin{:});
if ~isreal(scale)
    error('MAKESCALE:ScaleVal','SCALE must be a real number')
end

% Bound the scale
if scale < 1.1 || scale > 10
    warning('MAKESCALE:ScaleVal','SCALE has been capped to be between 1.1 and 10
for readability.')
end
scale = min(max(scale,1.1),10);
earthRadius = 6371000;

% Get the distance of the map
mlat = mean(latlim);
if abs(mlat) > 90;
    d = 0;
else
    d = earthRadius.*cosd(mlat).*deg2rad(diff(lonlim));
end
dmax = d/1.1;
dmin = d/10;
dlat = diff(latlim);
dlon = diff(lonlim);

% Calculate the distance of the scale bar
rnd2 = floor(log10(d/scale))-1;
dscale = round2(d/scale,10^rnd2);
```

```
% Cap it
if dscale > dmax;
    rnd2 = rnd2 - 1;
    dscale = round2(dmax,10^rnd2);
end
if dscale < dmin
    rnd2 = rnd2 - 1;
    dscale = round2(dmin,10^rnd2);
end

% Make the text string
if strcmpi(units,'si')
    if d > 1e3*scale
        dst = num2str(dscale/1e3);
        lbl = ' km';
    elseif d > scale
        dst = num2str(dscale);
        lbl = ' m';
    else
        dst = num2str(dscale*1e3);
        lbl = ' mm';
    end
else
    if d > scale/0.000621371192
        rnd2 = floor(log10(d/scale*0.000621371192))-1;
        dscale = round2(d/scale*0.000621371192,10^rnd2);
        dst = num2str(dscale);
        lbl = ' mi';
        dscale = dscale/0.000621371192;
    elseif d > scale*0.3048
        rnd2 = floor(log10(d/scale/0.3048))-1;
        dscale = round2(d/scale/0.30482,10^rnd2);
        dst = num2str(dscale);
        lbl = ' ft';
        dscale = dscale*.3048;
    else
        rnd2 = floor(log10(d/scale/0.3048*12))-1;
        dscale = round2(d/scale/0.30482*12,10^rnd2);
        dst = num2str(dscale);
        lbl = ' in';
        dscale = dscale/12*.3048;
    end
end

% Get the postions
d1 = [-0.02 0.05];
```

```
issouth = 0;
iseast = 0;
iswest = 0;
switch lower(location)
   case {'southeast','se'}
      issouth = 1;
      iseast = 1;
   case {'northeast','ne'}
      iseast = 1;
   case {'southwest','sw'}
      issouth = 1;
      iswest = 1;
   case {'northwest','nw'}
      iswest = 1;
   case {'north','n'}
   case {'south','s'}
      issouth = 1;
end

if issouth
   slat = latlim(1)+0.05*diff(latlim);
else
   slat = latlim(end)-0.08*diff(latlim);
end

if iseast
   slon = lonlim(end)-0.05*diff(lonlim);
   slon = [slon slon-rad2deg(dscale./(earthRadius.*cosd(mlat)))];
   slat = [slat slat];
elseif iswest
   slon = lonlim(1)+0.05*diff(lonlim);
   slon = [slon slon+rad2deg(dscale./(earthRadius.*cosd(mlat)))];
   slat = [slat slat];
   slon = fliplr(slon);
else
   slon = mean(lonlim);
   slon = slon + [-rad2deg(dscale./(earthRadius.*cosd(mlat)))/2
rad2deg(dscale./(earthRadius.*cosd(mlat))/2)];
   slat = [slat slat];
   slon = fliplr(slon);
end

% Get the box location
blat = [slat([2 1])+[d1(1)*dlat d1(2)*dlat] slat([1 2])+[d1(2)*dlat d1(1)*dlat]];
blat = blat([2:4 1]);
blon = [slon+[0.02*dlon -0.02*dlon] slon([2 1])+[-0.02*dlon 0.02*dlon]];
```

```
% Delete Old Scale
aold = gca;
axes(anum);
ch = get(anum,'Children');
isOldScale = strcmpi(get(ch,'Tag'),'MapScale');
delete(ch(isOldScale));

% Make the scale
washold = ishold;
hold on
hbox = patch(blon,blat,'w');
set(hbox,'Tag','MapScale');
hline = plot(slon,slat,'k','LineWidth',3);
set(hline,'Tag','MapScale');
units_axis = get(gca,'Units');
set(gca,'Units','Inches')
pos = get(gca,'OuterPosition');
sz = mean(pos(4));
htext = text(mean(blon),mean(blat)+.01*dlat,[dst
lbl],'HorizontalAlignment','center','FontSize',sz*2.3);
hzoom = zoom;
hpan = pan(gcf);
set(htext,'Tag','MapScale')
set(gca,'Units',units_axis);

% Set Resizer/Zoom/Pan/Click Callbacks
set(gcf,'ResizeFcn',{@ChangeTextSize,gca,htext});
set(hzoom,'ActionPostCallback',{@remakeZoomPanClick,anum,location,scale,units});
set(hpan,'ActionPostCallback',{@remakeZoomPanClick,anum,location,scale,units});
set(anum,'ButtonDownFcn',{@remakeZoomPanClick,anum,location,scale,units});
axes(aold);

% Output Handles
if nargout > 0
   h = [hbox; hline; htext];
end

% Restore Hold Off
if ~washold
   hold off
end

% Change the text font on figure resize.
function ChangeTextSize(~,~,anum,htext)
units = get(anum,'Units');
```

```
set(anum,'Units','Inches')
pos = get(anum,'OuterPosition');
sz = mean(pos(4));
set(htext,'FontSize',sz*2.3);
set(anum,'Units',units);

function remakeZoomPanClick(~,~,anum,location,scale,units)
makescale(anum,location,scale,'units',units);

function x = round2(x,base)
x = round(x./base).*base;

function [anum,latlim,lonlim,scale,location,units] = parseInputs(varargin)
% Default Values
anum = gca;
scale = 5;
location = 'se';
units = 'si';

% Loop through number of arguments in
ii = 1;
while ii <= length(varargin)

    % Either a axis number, or a scale value
    if isscalar(varargin{ii}) && isnumeric(varargin{ii})

        % Is it a non-root handle?
        if ishandle(varargin{ii}) && varargin{ii} ~= 0

            % Check if it is an axis number, not a figure number
            pos = get(varargin{ii},'ActivePositionProperty');
            if strcmpi(pos,'outerposition')
                anum = varargin{1};
                ii = ii + 1;
                continue;
            end
        end

        % Scale Value
        scale = varargin{ii};
        ii = ii + 1;

    % Locations
    elseif ischar(varargin{ii})
        if strcmpi(varargin{ii},'units')
            units = varargin{ii+1};
```

```
            ii = ii + 2;
        else
            locs = {'northeast','ne','north','n','southeast','se','south',...
                's','southwest','sw','northwest','nw'};
            if ~ismember(lower(varargin{ii}),locs)
                locOut = 'northeast, ne, north, n, southeast, se, south, s, southwest, sw, 
northwest, nw';
                error('MAKESCALE:LOCS',['LOCATION must be one of the following: ' 
locOut])
            end
            location = varargin{ii};
            ii = ii + 1;
        end
    end
end

% Get limits
latlim = get(anum,'YLim');
lonlim = get(anum,'XLim');


function varargout = plot_google_map(varargin)
% function h = plot_google_map(varargin)
% Plots a google map on the current axes using the Google Static Maps API
%
% USAGE:
% h = plot_google_map(Property, Value,...)
% Plots the map on the given axes. Used also if no output is specified
%
% Or:
% [lonVec latVec imag] = plot_google_map(Property, Value,...)
% Returns the map without plotting it
%
% PROPERTIES:
%    Height (640)   - Height of the image in pixels (max 640)
%    Width  (640)   - Width of the image in pixels (max 640)
%    Scale (2)      - (1/2) Resolution scale factor . using Scale=2 will
%                     double the resulotion of the downloaded image (up
%                     to 1280x1280) and will result in finer rendering,
%                     but processing time will be longer.
%    MapType        - ('roadmap') Type of map to return. Any of [roadmap,
%                     satellite, terrain, hybrid] See the Google Maps API for
%                     more information.
%    Alpha (1)      - (0-1) Transparency level of the map (0 is fully
%                     transparent). While the map is always
%                     moved to the bottom of the plot (i.e. will
%                     not hide previously drawn items), this can
```

```
%                  be useful in order to increase readability
%                  if many colors are ploted (using SCATTER
%                  for example).
%    ShowLabels (1) - (0/1) Controls wheter to display city/street textual labels on the
map
%    Language       - (string) A 2 letter ISO 639-1 language code for displaying labels in a
local language instead of English (where available).
%                  For example, for Chinese use:
%                  plot_google_map('language','zh')
%                  For the list of codes, see:
%                  http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
%    Marker         - The marker argument is a text string with fields
%                  conforming to the Google Maps API. The
%                  following are valid examples:
%                  '43.0738740,-70.713993' (default midsize orange marker)
%                  '43.0738740,-70.713993,blue' (midsize blue marker)
%                  '43.0738740,-70.713993,yellowa' (midsize yellow
%                  marker with label "A")
%                  '43.0738740,-70.713993,tinyredb' (tiny red marker
%                  with label "B")
%    Refresh (1)    - (0/1) defines whether to automatically refresh the
%                  map upon zoom/pan action on the figure.
%    AutoAxis (1)   - (0/1) defines whether to automatically adjust the axis
%                  of the plot to avoid the map being stretched.
%                  This will adjust the span to be correct
%                  according to the shape of the map axes.
%    APIKey         - (string) set your own API key which you obtained from Google:
%                  http://developers.google.com/maps/documentation/staticmaps/#api_key
%                  This will enable up to 25,000 map requests per day,
%                  compared to a few hundred requests without a key.
%                  To set the key, use:
%                  plot_google_map('APIKey','SomeLongStringObtaindFromGoogle')
%                  You need to do this only once to set the key.
%                  To disable the use of a key, use:
%                  plot_google_map('APIKey','')
%
% OUTPUT:
%    h             - Handle to the plotted map
%
%    lonVect       - Vector of Longidute coordinates (WGS84) of the image
%    latVect       - Vector of Latidute coordinates (WGS84) of the image
%    imag          - Image matrix (height,width,3) of the map
%
% EXAMPLE - plot a map showing some capitals in Europe:
%    lat = [48.8708   51.5188   41.9260   40.4312   52.523   37.982];
%    lon = [2.4131    -0.1300    12.4951   -3.6788    13.415   23.715];
```

```
%    plot(lon,lat,'.r','MarkerSize',20)
%    plot_google_map
%
% References:
% http://www.mathworks.com/matlabcentral/fileexchange/24113
% http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/
% http://developers.google.com/maps/documentation/staticmaps/
%
%  Acknowledgement to Val Schmidt for his submission of get_google_map.m
%
% Author:
%  Zohar Bar-Yehuda
%
% Version 1.4 - 25/03/2014
%       - Added the language parameter for showing labels in a local language
%       - Display the URL on error to allow easier debugging of API errors
% Version 1.3 - 06/10/2013
%       - Improved functionality of AutoAxis, which now handles any shape of map axes.
%         Now also updates the extent of the map if the figure is resized.
%       - Added the showLabels parameter which allows hiding the textual labels on the
map.
% Version 1.2 - 16/06/2012
%       - Support use of the "scale=2" parameter by default for finer rendering (set scale=1
if too slow).
%       - Auto-adjust axis extent so the map isn't stretched.
%       - Set and use an API key which enables a much higher usage volume per day.
% Version 1.1 - 25/08/2011

% store parameters in global variable (used for auto-refresh)
global inputParams
persistent apiKey
if isnumeric(apiKey)
    % first run, check if API key file exists
    if exist('api_key.mat','file')
        load api_key
    else
        apiKey = '';
    end
end
axHandle = gca;
inputParams.(['ax' num2str(axHandle*1e6,'%.0f')]) = varargin;

% Handle input arguments

height = 640;
width = 640;
```

```
scale = 2;
maptype = 'roadmap';
alphaData = 1;
autoRfresh = 1;
autoAxis = 1;
showLabels = 1;
language = '';
hold on
markeridx = 1;
markerlist = {};
if nargin >= 2
    for idx = 1:2:length(varargin)
        switch lower(varargin{idx})
            case 'height'
                height = varargin{idx+1};
            case 'width'
                width = varargin{idx+1};
            case 'maptype'
                maptype = varargin{idx+1};
            case 'alpha'
                alphaData = varargin{idx+1};
            case 'refresh'
                autoRfresh = varargin{idx+1};
            case 'showlabels'
                showLabels = varargin{idx+1};
            case 'language'
                language = varargin{idx+1};
            case 'marker'
                markerlist{markeridx} = varargin{idx+1};
                markeridx = markeridx + 1;
            case 'autoaxis'
                autoAxis = varargin{idx+1};
            case 'apikey'
                apiKey = varargin{idx+1}; % set new key
                % save key to file
                funcFile = which('plot_google_map.m');
                pth = fileparts(funcFile);
                keyFile = fullfile(pth,'api_key.mat');
                save(keyFile,'apiKey')
            otherwise
                error(['Unrecognized variable: ' varargin{idx}])
        end
    end
end
if height > 640
    height = 640;
```

```
end
if width > 640
    width = 640;
end

curAxis = axis;
% Enforce Latitude constraints of EPSG:900913
if curAxis(3) < -85
    curAxis(3) = -85;
end
if curAxis(4) > 85
    curAxis(4) = 85;
end
% Enforce longitude constrains
if curAxis(1) < -180
    curAxis(1) = -180;
end
if curAxis(1) > 180
    curAxis(1) = 0;
end
if curAxis(2) > 180
    curAxis(2) = 180;
end
if curAxis(2) < -180
    curAxis(2) = 0;
end

if isequal(curAxis,[0 1 0 1]) % probably an empty figure
    % display world map
    curAxis = [-200 200 -85 85];
    axis(curAxis)
end


if autoAxis
    % adjust current axis limit to avoid strectched maps
    [xExtent,yExtent] = latLonToMeters(curAxis(3:4), curAxis(1:2) );
    xExtent = diff(xExtent); % just the size of the span
    yExtent = diff(yExtent);
    % get axes aspect ratio
    drawnow
    org_units = get(axHandle,'Units');
    set(axHandle,'Units','Pixels')
    ax_position = get(axHandle,'position');
    set(axHandle,'Units',org_units)
    aspect_ratio = ax_position(4) / ax_position(3);
```

```
    if xExtent*aspect_ratio > yExtent
        centerX = mean(curAxis(1:2));
        centerY = mean(curAxis(3:4));
        spanX = (curAxis(2)-curAxis(1))/2;
        spanY = (curAxis(4)-curAxis(3))/2;

        % enlarge the Y extent
        spanY = spanY*xExtent*aspect_ratio/yExtent; % new span
        if spanY > 85
            spanX = spanX * 85 / spanY;
            spanY = spanY * 85 / spanY;
        end
        curAxis(1) = centerX-spanX;
        curAxis(2) = centerX+spanX;
        curAxis(3) = centerY-spanY;
        curAxis(4) = centerY+spanY;
    elseif yExtent > xExtent*aspect_ratio

        centerX = mean(curAxis(1:2));
        centerY = mean(curAxis(3:4));
        spanX = (curAxis(2)-curAxis(1))/2;
        spanY = (curAxis(4)-curAxis(3))/2;
        % enlarge the X extent
        spanX = spanX*yExtent/(xExtent*aspect_ratio); % new span
        if spanX > 180
            spanY = spanY * 180 / spanX;
            spanX = spanX * 180 / spanX;
        end

        curAxis(1) = centerX-spanX;
        curAxis(2) = centerX+spanX;
        curAxis(3) = centerY-spanY;
        curAxis(4) = centerY+spanY;
    end
    % Enforce Latitude constraints of EPSG:900913
    if curAxis(3) < -85
        curAxis(3:4) = curAxis(3:4) + (-85 - curAxis(3));
    end
    if curAxis(4) > 85
        curAxis(3:4) = curAxis(3:4) + (85 - curAxis(4));
    end
    axis(curAxis) % update axis as quickly as possible, before downloading new image
    drawnow
end
```

```
% Delete previous map from plot (if exists)
if nargout <= 1 % only if in plotting mode
   curChildren = get(axHandle,'children');
   map_objs = findobj(curChildren,'tag','gmap');
   bd_callback = [];
   for idx = 1:length(map_objs)
      if ~isempty(get(map_objs(idx),'ButtonDownFcn'))
         % copy callback properties from current map
         bd_callback = get(map_objs(idx),'ButtonDownFcn');
      end
   end
   delete(map_objs)

end

% Calculate zoom level for current axis limits
[xExtent,yExtent] = latLonToMeters(curAxis(3:4), curAxis(1:2) );
minResX = diff(xExtent) / width;
minResY = diff(yExtent) / height;
minRes = max([minResX minResY]);
tileSize = 256;
initialResolution = 2 * pi * 6378137 / tileSize; % 156543.03392804062 for tileSize 256
pixels
zoomlevel = floor(log2(initialResolution/minRes));

% Enforce valid zoom levels
if zoomlevel < 0
   zoomlevel = 0;
end
if zoomlevel > 19
   zoomlevel = 19;
end

% Calculate center coordinate in WGS1984
lat = (curAxis(3)+curAxis(4))/2;
lon = (curAxis(1)+curAxis(2))/2;

% CONSTRUCT QUERY URL
preamble = 'http://maps.googleapis.com/maps/api/staticmap';
location = ['?center=' num2str(lat,10) ',' num2str(lon,10)];
zoomStr = ['&zoom=' num2str(zoomlevel)];
sizeStr = ['&scale=' num2str(scale) '&size=' num2str(width) 'x' num2str(height)];
maptypeStr = ['&maptype=' maptype ];
if ~isempty(apiKey)
   keyStr = ['&key=' apiKey];
else
```

```
    keyStr = '';
end
markers = '&markers=';
for idx = 1:length(markerlist)
    if idx < length(markerlist)
        markers = [markers markerlist{idx} '%7C'];
    else
        markers = [markers markerlist{idx}];
    end
end
if showLabels == 0
    labelsStr = '&style=feature:all|element:labels|visibility:off';
else
    labelsStr = '';
end
if ~isempty(language)
    languageStr = ['&language=' language];
else
    languageStr = '';
end

if ismember(maptype,{'satellite','hybrid'})
    filename = 'tmp.jpg';
    format = '&format=jpg';
    convertNeeded = 0;
else
    filename = 'tmp.png';
    format = '&format=png';
    convertNeeded = 1;
end
sensor = '&sensor=false';
url = [preamble location zoomStr sizeStr maptypeStr format markers labelsStr
languageStr sensor keyStr];

% Get the image
try
    urlwrite(url,filename);
catch % error downloading map
    warning(sprintf(['Unable to download map form Google Servers.\n' ...
        'Possible reasons: no network connection, quota exceeded, or some other error.\n' ...
        'Consider using an API key if quota problems persist.\n\n' ...
        'To debug, try pasting the following URL in your browser, which may result in a
more informative error:\n%s'], url));
    varargout{1} = [];
    varargout{2} = [];
    varargout{3} = [];
```

```
    return
end
[M Mcolor] = imread(filename);
M = cast(M,'double');
delete(filename); % delete temp file
width = size(M,2);
height = size(M,1);

% Calculate a meshgrid of pixel coordinates in EPSG:900913
centerPixelY = round(height/2);
centerPixelX = round(width/2);
[centerX,centerY] = latLonToMeters(lat, lon ); % center coordinates in EPSG:900913
curResolution = initialResolution / 2^zoomlevel/scale; % meters/pixel (EPSG:900913)
xVec = centerX + ((1:width)-centerPixelX) * curResolution; % x vector
yVec = centerY + ((height:-1:1)-centerPixelY) * curResolution; % y vector
[xMesh,yMesh] = meshgrid(xVec,yVec); % construct meshgrid

% convert meshgrid to WGS1984
[lonMesh,latMesh] = metersToLatLon(xMesh,yMesh);

% We now want to convert the image from a colormap image with an uneven
% mesh grid, into an RGB truecolor image with a uniform grid.
% This would enable displaying it with IMAGE, instead of PCOLOR.
% Advantages are:
% 1) faster rendering
% 2) makes it possible to display together with other colormap annotations (PCOLOR,
SCATTER etc.)

% Convert image from colormap type to RGB truecolor (if PNG is used)
if convertNeeded
    imag = zeros(height,width,3);
    for idx = 1:3
        imag(:,:,idx) = reshape(Mcolor(M(:)+1+(idx-1)*size(Mcolor,1)),height,width);
    end
else
    imag = M/255;
end

% Next, project the data into a uniform WGS1984 grid
sizeFactor = 1; % factoring of new image
uniHeight = round(height*sizeFactor);
uniWidth = round(width*sizeFactor);
latVect = linspace(latMesh(1,1),latMesh(end,1),uniHeight);
lonVect = linspace(lonMesh(1,1),lonMesh(1,end),uniWidth);
[uniLonMesh,uniLatMesh] = meshgrid(lonVect,latVect);
uniImag = zeros(uniHeight,uniWidth,3);
```

```
% old version (projection using INTERP2)
% for idx = 1:3
%     % 'nearest' method is the fastest. difference from other methods is neglible
%         uniImag(:,:,idx) =
interp2(lonMesh,latMesh,imag(:,:,idx),uniLonMesh,uniLatMesh,'nearest');
% end
uniImag =  myTurboInterp2(lonMesh,latMesh,imag,uniLonMesh,uniLatMesh);

if nargout <= 1 % plot map
   % display image
   h = image(lonVect,latVect,uniImag);
   set(gca,'YDir','Normal')
   set(h,'tag','gmap')
   set(h,'AlphaData',alphaData)

   % add a dummy image to allow pan/zoom out to x2 of the image extent
   h_tmp = image(lonVect([1 end]),latVect([1 end]),zeros(2),'Visible','off');
   set(h_tmp,'tag','gmap')

   % older version (display without conversion to uniform grid)
   % h =pcolor(lonMesh,latMesh,(M));
   % colormap(Mcolor)
   % caxis([0 255])
   % warning off % to avoid strange rendering warnings
   % shading flat

   uistack(h,'bottom') % move map to bottom (so it doesn't hide previously drawn
annotations)
   axis(curAxis) % restore original zoom
   if nargout == 1
      varargout{1} = h;
   end
   % if auto-refresh mode - override zoom callback to allow autumatic
   % refresh of map upon zoom actions.
   zoomHandle = zoom;
   panHandle = pan;
   if autoRferesh
      set(zoomHandle,'ActionPostCallback',@update_google_map);
      set(panHandle, 'ActionPostCallback', @update_google_map);
   else % disable zoom override
      set(zoomHandle,'ActionPostCallback',[]);
      set(panHandle, 'ActionPostCallback',[]);
   end

   % set callback for figure resize function, to update extents if figure
```

```
    % is streched.
    figHandle = get(axHandle,'Parent');
    set(figHandle, 'ResizeFcn', @update_google_map_fig);

    % set callback properties
    set(h,'ButtonDownFcn',bd_callback);
else % don't plot, only return map
    varargout{1} = lonVect;
    varargout{2} = latVect;
    varargout{3} = uniImag;
end

% Coordinate transformation functions

function [lon,lat] = metersToLatLon(x,y)
% Converts XY point from Spherical Mercator EPSG:900913 to lat/lon in WGS84
Datum
originShift = 2 * pi * 6378137 / 2.0; % 20037508.342789244
lon = (x ./ originShift) * 180;
lat = (y ./ originShift) * 180;
lat = 180 / pi * (2 * atan( exp( lat * pi / 180)) - pi / 2);

function [x,y] = latLonToMeters(lat, lon )
% Converts given lat/lon in WGS84 Datum to XY in Spherical Mercator EPSG:900913"
originShift = 2 * pi * 6378137 / 2.0; % 20037508.342789244
x = lon * originShift / 180;
y = log(tan((90 + lat) * pi / 360 )) / (pi / 180);
y = y * originShift / 180;

function ZI = myTurboInterp2(X,Y,Z,XI,YI)
% An extremely fast nearest neighbour 2D interpolation, assuming both input
% and output grids consist only of squares, meaning:
% - uniform X for each column
% - uniform Y for each row
XI = XI(1,:);
X = X(1,:);
YI = YI(:,1);
Y = Y(:,1);

xiPos = nan*ones(size(XI));
xLen = length(X);
yiPos = nan*ones(size(YI));
yLen = length(Y);
% find x conversion
xPos = 1;
for idx = 1:length(xiPos)
```

```
    if XI(idx) >= X(1) && XI(idx) <= X(end)
        while xPos < xLen && X(xPos+1)<XI(idx)
            xPos = xPos + 1;
        end
        diffs = abs(X(xPos:xPos+1)-XI(idx));
        if diffs(1) < diffs(2)
            xiPos(idx) = xPos;
        else
            xiPos(idx) = xPos + 1;
        end
    end
end
% find y conversion
yPos = 1;
for idx = 1:length(yiPos)
    if YI(idx) <= Y(1) && YI(idx) >= Y(end)
        while yPos < yLen && Y(yPos+1)>YI(idx)
            yPos = yPos + 1;
        end
        diffs = abs(Y(yPos:yPos+1)-YI(idx));
        if diffs(1) < diffs(2)
            yiPos(idx) = yPos;
        else
            yiPos(idx) = yPos + 1;
        end
    end
end
ZI = Z(yiPos,xiPos,:);

function update_google_map(obj,evd)
% callback function for auto-refresh
drawnow;
global inputParams
if isfield(inputParams,['ax' num2str(gca*1e6,'%.0f')])
    params = inputParams.(['ax' num2str(gca*1e6,'%.0f')]);
    plot_google_map(params{:});
end

function update_google_map_fig(obj,evd)
% callback function for auto-refresh
drawnow;
global inputParams
axes_objs = findobj(get(gcf,'children'),'type','axes');
for idx = 1:length(axes_objs)
    if ~isempty(findobj(get(axes_objs(idx),'children'),'tag','gmap'));
        if isfield(inputParams,['ax' num2str(axes_objs(idx)*1e6,'%.0f')])
```

```
            params = inputParams.(['ax' num2str(axes_objs(idx)*1e6,'%.0f')]);
        else
            params = {};
        end
        axes(axes_objs(idx));
        plot_google_map(params{:});
        break;
    end
end
```