

2017

Microseismic Event Characteristics Associated With The Spatial And Temporal Stages of Hydraulic Fracture Growth in Laboratory Experiments

Abigail Arielle Maxwell
Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses



Part of the [Earth Sciences Commons](#)

Recommended Citation

Maxwell, Abigail Arielle, "Microseismic Event Characteristics Associated With The Spatial And Temporal Stages of Hydraulic Fracture Growth in Laboratory Experiments" (2017). *LSU Master's Theses*. 4562.
https://digitalcommons.lsu.edu/gradschool_theses/4562

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

MICROSEISMIC EVENT CHARACTERISTICS ASSOCIATED WITH THE SPATIAL AND
TEMPORAL STAGES OF HYDRAULIC FRACTURE GROWTH IN LABORATORY
EXPERIMENTS

A Thesis

Submitted to the Graduate Faculty of the

Louisiana State University and

Agricultural and Mechanical College

in partial fulfilment of the

requirements for the degree of

Master of Science

in

The Department of Geology and Geophysics

by

Abigail Arielle Maxwell

B.Sc., University of Georgia, 2013

May 2017

Acknowledgments

I would like to express my sincerest appreciation to my advisor, Dr. Juan Lorenzo, for his patience, and dedicated mentorship throughout the course of this research. Special thanks to my co-advisor, Dr. Arash Dahi-Taleghani and committee member, Dr. Karen Luttrell for their time and expert advice that helped me through this journey. I would also like to thank my family and friends for their constant encouragement. Very special thanks to my fiancé for being a constant pillar of strength for me. I would like to also specially acknowledge my parents for being my biggest cheerleaders. Thank you to the student worker in our lab, Jack Cadigan, LSU Machine Shop, and Dr. Lorenzo's research group, for helping me to survive graduate school life. I would like to thank the LSU Department of Geology and Geophysics for funding me for the entire duration of my stay and also for the Departmental scholarship. Thank you to all the organizations that helped to fund this project and my stay- Geological Society of America Student Research Grant, Geological Society of America Geophysics Student Travel Grant for GSA, Society of Petroleum Engineers Delta Chapter Student Scholarship, Shreveport Geological Society Scholarship, and the National Association of Black Geoscientists Scholarship.

Table of Contents

Acknowledgments.....	ii
List of Figures	v
List of Tables	xiv
Abstract.....	xvi
CHAPTER	
1 Introduction.....	Error! Bookmark not defined.
1.1 Problems and Objectives.....	1
1.2 Microseismicity associated with hydraulic fracturing.....	2
1.3 Seismicity related to magmatic fracture growth	3
1.4 Comparison of the seismicity in hydraulic and magmatic fracturing	4
1.5 Key Concepts	14
1.5.1 Fluid-driven fractures.....	14
1.5.2 Fluid-driven fracture mechanics	14
1.6 Application of the proposed seismic model	17
1.6.1 Hydraulic fracturing process	18
1.6.2 Microseismic monitoring during hydraulic fracturing in the industry.....	19
2 Methods.....	21
2.1 Experimental Set-up.....	21
2.2 Laboratory Scaling	21
2.3 Experimental Procedure.....	26
2.4 Seismic data processing	32
2.4.1 Microseismic event selection and Event characteristics	32
2.4.2 Location of Events.....	36
2.5 Pressure Data Analysis	47
2.6 Camera Data.....	49
3 Results and Interpretations	51
3.1 Microseismic event classifications	51
3.2 Microseismic event locations.....	60
3.3 Fracture dimensions.....	71

3.3.1	Fracture Shape	71
3.3.2	Fracture surface	74
3.3.3	Fracture dimensions	75
3.4	Pressure response	83
3.4.1	Pressure Variations	83
3.5	Relating the pressure response, seismic event occurrence and fracture growth	86
4	Discussion	91
4.1	Comparing laboratory seismic data to seismic data in the field.....	91
4.2	The microseismic event variations in experiment 1 and 2	92
4.3	The pressure variations in experiment 1 and 2	95
4.4	Future Work	96
5	Conclusions.....	98
	References	100
	Appendix	106
A.	Non-dimensional toughness parameters	106
B.	Sample preparation procedure.....	107
C.	Fracturing Fluid Preparation	113
D.	Experimental Procedure	115
E.	Characteristics and locations for the microseismic events in Experiment 1.....	118
F.	Characteristics and locations for the the microsesimic events in Experiment 2.....	119
G.	Locations of the microseismic events.....	124
H.	Seismic, pump pressure, and camera data in reference time frame Experiment 1...126	
I.	Seismic, pump pressure, and camera data in reference time frame Experiment 2...127	
J.	Programs.....	128
	Vita	333

List of Figures

Figure 1.1. Low frequency and high frequency events from hydraulic fracturing treatment in the Montney Field in north-eastern Canada (Eaton et al., 2013). a) The frequency spectrum of the two events. The LFE corresponds to frequencies with a maximum of 60 Hz while HFE corresponds to maximum of ~ 110 Hz. b) The signal (measured in velocity) of the low frequency event followed by the high frequency event.	3
Figure 1.2. An example of volcanic tremor. These events are characterized by frequencies of less than 1 Hz for a duration lasting from a minute to months (Wasserman, 2002).	4
Figure 1.3. Material Point Method modelling for the variation in stresses as a hydraulic fracture approaches a pre-existing fracture at an angle of 60° , with a pressure of 10 MPa applied to the hydraulic fracture (Aimene and Nairn 2014). The two horizontal stresses are equal i.e. anisotropy, measured as the ratio of the two horizontal stresses, is equal to 1. At NF position 1 the hydraulic fracture is closer to the pre-existing fracture than NF position 2. a) shows the variation of stress in the x direction while b) shows the variation of stress in the x direction. The blue color shows the tensile stresses and red shows compressional stresses. .	8
Figure 1.4. Extended finite element modelling results of the debonding that takes places as a fluid-driven fracture approaches a non-orthogonal pre-existing sealed fracture (Dahi Taleghani 2009) (a) A fluid-driven fracture (blue) approaches a pre-existing sealed fracture (grey). (b) The debonded zone (blue along the pre-existing fracture (grey) is asymmetric and thus can result in the formation of a one direction deflection along the pre-existing fracture when the two fractures intersect.	9
Figure 1.5. The possible scenarios when a fluid-driven fracture intersects a natural fracture (Warpinski and Teufel 1987, Dahi Taleghani 2009).	10
Figure 1.6. Block diagrams showing the three modes of crack surface displacements. Mode I (a) illustrates a tensile-mode crack displacement which is the equivalent of opening, Mode II (b) illustrates a shear-mode crack displacement which is in plane shearing, and Mode III (c) illustrates a shear-mode crack displacement which is out-of-plane shearing. (van der Pluijm and Marshak 2004).	15
Figure 1.7. Composite Griffith-Coulomb failure envelope shown on a Mohr diagram for an intact, homogenous, isotropic rock with a tensile strength, T. The Mohr diagram shows shear stress, τ , versus effective normal stress ($\sigma_n - P_f$, where σ_n , is the normal stress and P_f , fluid-pressure in fracture). σ_1 is the maximum principal stress, σ_3 is the minimum principal stress, μ_i is the coefficient of friction along a fault and Θ_s is the angle between the shear fracture and σ_1 . The criteria for different failure modes (i) extensional (mode I), (ii) extensional-shear, and (iii) compressional shear failure for a rock are shown (mode II and mode III) (adapted from Sibson, 2000).	16

Figure 1.8. A schematic showing the steps in hydraulic fracturing. Hydraulic fracturing fluid is pumped into the target formation via deviated well (the well is initially vertical but is deviated so that it is horizontal within the target formation). The high fluid pressure results in the formation of hydraulic fractures which are controlled by maintaining a specific treatment pressure and monitoring the growth through microseismic monitoring (Zuppann and Steinmetz 2014).19

Figure 2.1. Dimensionless toughness for three materials- polycarbonate, polymethyl methacrylate (PMMA) and shale. We calculate the dimensionless toughness using equation 2.1 and inserting the appropriate values for each material and the parameters for the experiment (Appendix). PMMA is the material we choose to use as our samples because the experiment using PMMA and the conditions we set for the laboratory has a dimensionless toughness of approximately 1 which means that we expect the PMMA to deform similarly to shale in the hydraulic fracturing laboratory experiments. Additionally, PMMA is cheap, readily available and transparent. Since it is an isotropic, homogenous medium, we use PMMA as a simple case for the generation of our model.25

Figure 2.2. Sample 1 (without the fault) in top view (a) and side view (b) showing the stations on top of the sample labelled A-D (the four stations on the top of sample 1) with the three piezo-electric sensors attached to each station in a Galperin arrangement. Each sensor is connected to a channel that is part of the seismic acquisition system. The well, of diameter $\frac{3}{4}$ " is in the center of the sample (the pink fluid is the fracturing fluid used to fill the well before the experiment begins.27

Figure 2.3. Sample 2 (with the pre-existing fracture) in top view. Sample 2 consists of 2 PMMA blocks each with one side pre-cut to fit together at an angle 60° . The two blocks are held together by JB-Weld (glue). The yellow dashed line represents the pre-existing fracture within sample 2. The stations on the top of the sample are labelled A-D with the three piezo-electric sensors attached to each station in a Galperin arrangement (Galperin 1955). Each sensor is connected to a channel that is part of the seismic acquisition system. The well, of diameter 0.019 m is also filled with the fracturing fluid before the start of the experiment.28

Figure 2.4. Diagram of the biaxial press containing the sample. The green shaded blocks represent the PMMA blocks that help to hold the sample in place during the experiment. The sample (in white) is in the middle of the press with 4 sensors seen on the top of the sample. We adjust the four steel plates in the middle of the press to secure the sample in the press. We set the pistons and measure the pressure using the gauges to ensure that the press exerts 1000 psi of pressure in the two horizontal directions.30

Figure 2.5. Flowchart of the steps in seismic processing: (1) We pick the seismic events above a minimum value of displacement of 80 counts and then, we keep events whose displacement we can pick above the noise level. (2) We cross correlate the data to align events since each acquisition card has an independent clock. (3) We calculate the signal-to-noise ratio and dominant frequency of each event. (4) Using the event characteristics from (3), duration, whether the event occurs singly or in a group, we classify the events. (5) We find the location

of the events by using principal component analysis and a back projection to the source of the seismic events, compare the locations with the camera images and identify where each type of event occurs. (6) We calculate the magnitude of each event. The programs in step 3,5 &6 are in the Appendix.34

Figure 2.6. Illustration of the particle motion of the displacement that occurs during a hypothetical seismic event (thin blue line). The best fit polarization of the particle motion has the eigenvectors ($V1$, $V2$, $V3$) as its axes (represented in red) with the longest axis representing $V1$. The azimuth, ϕ , and dip, Θ , are illustrated by the double-arrowed curved lines (Saenger, Schmalholz et al. 2009).38

Figure 2.7. Principal component analysis of Event 1 from experiment 2 (Figure 3.1). The red circles represent the maximum value of the energy of the event (b), the corresponding azimuth and dip associated with the maximum energy (c and d respectively). We observe the variability of the azimuth and dip of the direction of motion of the seismic waves which is due to the influence of the noise.....39

Figure 2.8. Magnification window of the maximum energy (black) for the event showing azimuth (blue) and dip (green) for the given window as well. Angle 1 for the azimuth is 290 and angle 1 for the dip is 86 (where the dotted red line intersects the blue and green lines respectively). Within this window, we find the standard deviation for the azimuth and dip and the average azimuth and dip angles.40

Figure 2.9. Diagram of an SVD test of 8 lines with known equations. The black lines within the box represent the input equations, the red squares represent the station locations in this example, and the blue square represents of solution of the SVD technique which correctly gives the intersection point of the 8 lines used.....42

Figure 2.10. Illustrations of the steps in refinement of all of the possible locations for event 1 in experiment 2 using the SVD technique, kernel density function and error ellipsoid fitting. (1) shows the results of the SVD for all the combinations of the locations (1024) for each event on a schematic diagram of sample 2. (2) is a plot of the kernel density function for the x coordinates of all the possible locations. The peak in the graph corresponds to the coordinate with the greatest number of locations closest to it. (3a) shows the possible locations left (< 100) after using the kernel density function on x, y and z coordinates. (3b) shows an error ellipsoid fit to the remaining locations. The center gives the actual location we use and the axes of the ellipsoid to give us the errors in the location.....46

Figure 2.11. Image of the crack during experiment 1 at 146.47 minutes from the start of pumping at the beginning of the experiment. Length 1, length 2, the area and the perimeter of the crack are the measurements we take at every 2 to 3 minutes of recording from the beginning to the end of fracture growth.50

Figure 2.12. Image of the created fracture during experiment 2 at 268.29 minutes from the start of pumping at the beginning of the experiment. At approximately 138.63 minutes into

the experiment, the field of view of the camera changes so that the eastern side of the created fracture is not completely visible. We estimate the size of the area cut off for the measurements of the fracture.....50

Figure 3.1. Microseismic event type I at station A, occurs as a single arrival with a frequency > 10,000 Hz lasting 0.5 seconds. The components, X, Y, and Z are in the principal coordinate system of north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.52

Figure 3.2. Microseismic event type II-a at station A occurs as a cluster of 2 events with an average dominant frequency > 10,000 Hz, lasting 0.5 seconds. The three components, X, Y and Z are in the principal coordinate system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.53

Figure 3.3. Microseismic event type II-b at station A with a frequency > 10,000 Hz occurs in a cluster with 5 identifiable arrivals. The 3 components X, Y, and Z, are in the principal system north, east and up.....54

Figure 3.4. Microseismic event type III at station A with a frequency < 10,000 Hz occurs as a single event. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.55

Figure 3.5. Microseismic event type IV-a at station A with a frequency < 10,000 Hz occurs as a cluster of 2 events. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per second.....56

Figure 3.6. Microseismic event type IV-b at station A with a frequency < 10,000 Hz occurs as a cluster with more than 2 events. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per second.....57

Figure 3.7. The distribution of the total number of microseismic events and the number of each event type throughout experiment 2. Most events occur between 5,000 and 20,000 seconds (during fracture growth), but we identify events that occur before fracture initiation. The events vary from low frequency to high frequency with no quantifiable pattern observed between the occurrence of event types. Event type I has the highest number of events throughout the experiment. Right before the intersection of the growing fracture with the pre-existing fracture at 16097 seconds, there is a rapid increase in the number of events occurring from less than 5 to 21 events in less than 10 minutes.....59

Figure 3.8. Locations of the events detected for experiments 1 & 2 with error ellipses, without the error ellipses and color coded based on the time of the occurrence of the microseismic event. (a) & (b) For experiment 1, most of the events occur in the vicinity of the well bore or within the body of the fracture close to the wellbore. (c) & (d) For experiment 2, the events identified occur at different locations throughout the fracture, near the wellbore and also in close association to the pre-existing fracture. (Appendix for 2D plots of locations)62

Figure 3.9. The locations of the events from experiment 2 in relation to the timing of their occurrence (Locations of the microseismic events in experiment 2 in Appendix F). The majority of events occur in relation to the fault (a) and ahead of the fracture (c). There is a general increase in the number of events as the fracture approaches the fault and intersects it (between 16,000 and 17, 000 seconds). The blue dotted line is at fracture initiation and the brown dotted line is at the intersection of the growing fracture with the pre-existing fracture.63

Figure 3.10 a) The location of Event 1 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline.64

Figure 3.11 a) The location of Event 24 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 24 occurs ahead of the fracture and before the pre-existing fracture.65

Figure 3.12. a) The location of Event 38 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 38 occurs on the pre-existing fracture as the fracture is still growing and has not yet intersected the pre-existing fracture.66

Figure 3.13. a) The location of Event 82 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 82 occurs at the tip of the fracture.....67

Figure 3.14. a) The location of Event 98 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 98 occurs on the other side of the pre-existing fault.....68

Figure 3.15. a) The location of Event 1 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 104 occurs within the body of the fracture.69

Figure 3.16. Top view sample 1 showing the fracture created in experiment 1. The pink fluid shows the extent of the fracture. The edge of the fracture itself is not fully rounded and smooth.72

Figure 3.17. The relationship between length 1 and length 2 (as defined in Figure 2.11) of the created fracture in experiment 1. The slope of the graph shows a 1:1 relationship between the lengths verifying that the fracture grows radially throughout the course of the experiment.....72

Figure 3.18. Top view of sample 2 showing the fracture created. The pink fluid shows the extent of the fracture. The created fracture intersects the pre-existing fracture (model fault) and is diverted into the fracture (pink semi-circle along the fault as labelled).73

Figure 3.19. The relationship between length 1 and length 2 (as defined in Figure 2.12) for experiment 2. The slope of the graph shows that length 2 grows at a slower rate as compared to length 1 during the experiment. This indicates that during the experiment length 2 is smaller than length 1 suggesting that fracture growth is not radial in experiment 2.73

Figure 3.20. Side views of the sample from experiment 1 (a) and the sample from experiment 2 (b). In (a), there are two linear features that show vertical displacement on the fracture surface (blue arrows) and rib-marks (red arrows). In (b), we see irregularities on the surface of the fracture as well (blue arrows). The fractures in both samples show some evidence of twisting in the fracture growth rather a perfectly horizontal fracture surface.75

Figure 3.21. For experiment 1, fracture growth measurements, Length 1 and Length 2 (as identified in methods showing the measurement of the fractures using ImageJ software), the area, and the perimeter of fractures. The error in measurement for the values including human error in measurement and ImageJ errors is ± 0.05 cm.77

Figure 3.22. For experiment 2, fracture growth measurements, Length 1 and Length 2 (as identified in methods showing the measurement of the fractures using ImageJ software), the area, and the perimeter of fractures. The error in measurement for the values from human error in measurement and ImageJ errors is ± 0.05 cm. The dashed lines shown on the plots represent time that the camera position changes and the crack is not fully seen in the recordings after this point. We add an estimate of the missing section of the crack to the measurements of the crack that we are able to obtain from the video recordings.78

Figure 3.23. The lengths (1 and 2), area, perimeter, and average width of the created fracture during the experiment 1. As there is a rise in the lengths, area and perimeter of the fracture, there is a decrease in the width generally. There are two points where there is a small increase and then a rapid decrease in width size occurring at $\sim 8,000$ seconds and $10,500$ seconds. .79

Figure 3.24. The lengths (1 and 2), area, perimeter, and average width of the created fracture during the experiment 1. As there is an increase in the lengths, area and perimeter of the fracture, there is also a general increase in the width size (Figure 3.18). There are fluctuations as well in the width size similar to fluctuations observed in the pump pressure response. The dashed line represents the time in the experiment where the camera changes orientation.80

Figure 3.25. The variation of fracture width during experiment 2 and pump pressure response. The width initially decreases rapidly at the start of fracture growth similarly to the decrease in the pump pressure before $4,800$ seconds after which an inverse relationship is observed where there is a decrease in the pressure response and there is a peak in the width size.81

Figure 3.26. The number of seismic events for 5 minute intervals in relation to the rate of growth of the crack's radius calculated from the area ($r = \sqrt{(A/\pi)}$) and the pump pressure response for experiment 1. The error in measurements are ± 0.9 cm for the area.87

Figure 3.27. Pump pressure data, microseismic event occurrence with respect to time and event amplitude for experiment 1. The events with the greatest amplitudes correspond to the timing of the drops in the pressure response. The microseismic event with the greatest maximum amplitude (amplitude is on a log scale) occurs right after the first pressure drop near the onset of fracture growth.88

Figure 3.28. The number of seismic events in 5 minute intervals, the rate of growth of the crack's radius calculated from the area ($r = \sqrt{(A/\pi)}$) and the pump pressure for experiment 2. The error in measurements are ± 0.9 cm for the area. The dotted brown line indicates the intersection of the growing fracture with the pre-existing fracture.89

Figure 3.29. Pump pressure data, microseismic event occurrence with respect to time and event amplitude for experiment 2. The events with the greatest amplitudes do not necessarily correspond to the timing of the drops in the pressure response as observed in experiment 1. The microseismic event with the greatest maximum amplitude (amplitude is on log scale) occurs right before the first pressure drop and before the onset of fracture growth. The events observed before fracture growth when located are related to the pre-existing fault and the borehole with the largest amplitude event before the pressure drop located on the fault.90

Figure B.1. The PMMA blocks during the sample preparation. Two blocks on the left of the figure already contain the wells and pre-cracks. Two blocks on the right of the figure show

one side each cut at angle of 60° from the horizontal before the two blocks are bound to create sample 2.....108

Figure B.2. Preparation of sample 2. a) The finished product, Sample two, the glue completely dries and sets. b) JB-Weld mixture that is applied to the separate blocks in order to bind the blocks. After the glue is applied, vices hold the sample together to make sure that the two blocks stay in place while the glue dries.....108

Figure B.3. The interference figures of sample 1 after the drilling of the well and thermal annealing (a) and after the valve has been added to the wellbore (b). These pictures show that after rigorous thermal annealing, the induced stresses are not fully removed from the samples.110

Figure B.4. Sample 1 oriented with its top to the right of the picture. 0.0191 m diameter borehole is filled with fracturing fluid (pink) and capped with valve to which the pump is attached for the experiment. The pre-crack is visible approximately at half of the sample's height (~ 0.038 m).....110

Figure B.5. (a) A 12-Channel amplifier and power supply (b) Set up of each station on the samples. We attach four sensor mounts to the top of the sample and four to the bottom of the sample. Each sensor mount has three holders for the sensors. We attach the sensors to each station and a cable, which acts as a channel connecting each sensor to the acquisition system, to each sensor.111

Figure B.6. Inline pressure sensor (Industrial Pressure Transducer Model 522 from Setra) that is attached to the top of the well.112

Figure D.1. Biaxial press where the sample is in the middle of the four steel plates. 18" tubing connects the pump to the well to direct the flow of the fracturing fluid from the pump to the well.....115

Figure D.2. Still image of the fracture during experiment 2 showing the outline of the fracture (yellow line) and the red LED light on during a synchronization signal.....117

Figure D.3. Schematics of the acquisition system for the experiments117

Figure G.1. X, Y and Z locations for the microseismic events occurring in experiment 1. The colors correspond to the timing of the microseismic events. (a) X and Z locations of the microseismic events. (b) Y and Z locations of the microseismic events.....124

Figure G.2. X, Y and Z locations for the microseismic events occurring in experiment 2. The colors correspond to the timing of the microseismic events. (a) X and Z locations of the microseismic events. (b) Y and Z locations of the microseismic events.....125

Figure H.1. The pump pressure response during the course of experiment 1 showing where the seismic and camera data were collected relative to the reference time frame for the

experiment with time 0 at the start of the experiment. The pressure gradually increases until the breakdown pressure is reached (~ 810 psi), where it levels and then rapidly declines until the pressure reaches approximately 450 psi. The pressure decline is gradual with two main drops in the pressure both ~ 50 psi. Overall the pressure declines to 350 psi at the end of the experiment. Experiment 1 contains three types of events- high frequency cluster, high frequency single event and one low frequency cluster. These events mainly occur at or along the rapid drops in pressure mentioned with only 2 of the 16 identified events occurring on the gentler slope of the decline.....126

Figure I.1. The pump pressure response during the course of experiment 2 showing where the seismic and camera data were collected relative to the reference time frame for the experiment with time 0 at the start of the experiment. The pressure gradually increases until the breakdown pressure is reached (~ 1600 psi), where it rapidly declines until the pressure reaches approximately 650 psi. The overall pressure decline is gradual after the first main pressure drop, however the pressure oscillates as it declines. Overall, the pressure declines to 600 psi at the end of the experiment but with short increases and decreases. Experiment 2 contains four main types of events- high frequency cluster, high frequency single event and low frequency clusters and low frequency single events. These events mainly occur at or along the increasing and decreasing slopes of the oscillations as well as close to the maxima and minima of the oscillations.....127

Figure J.1. Flowchart showing the order that the programs should be used when pre-processing the raw data recorded from the piezoelectric sensors.129

Figure J.2. Flowchart showing the programs that we use to obtain the event characteristics for the categorization of events and mapping of each event type within each sample.....130

List of Tables

Table 1.1. Summary of the seismic events related to magmatic fracturing and hydraulic fracturing in the oil and gas industry grouped by their frequency types. (¹ Maxwell 2011; ² Eaton et al., 2013; ³ Tary et al., 2014; ⁴ Wasserman, 2002; ⁵ Zobin, 2012)	6
Table 1.2. A proposed model showing the probable origins and locations of the seismic events occurring in association with fluid-driven fracture growth. The origins and locations are likely based on the assumptions that particular mechanisms at different stages of the fracture growth will produce seismic events of different characteristics such as frequency or amplitude (¹ Aimene and Nairn, 2014; ² Chouet, 1986; ³ Dahi Taleghani and Olson, 2014; ⁴ Eaton et al, 2013; ⁵ Frash, 2007; ⁶ Maxwell, 2014; ⁷ Maxwell et al., 2015, ⁸ Rutledge et al., 2015; ⁹ Shelly et al., 2007; ¹⁰ Tary et al., 2014; ¹¹ Wasserman, 2002; ¹² Wu et. al, 2007; ¹³ Zobin, 2012).	13
Table 2.1. The dimensions of sample 1 and sample 2 used in the experiments (length, width and heig/ht of each sample). The error in measurements is +/- 0.0005 m	21
Table 2.2. Properties of polymethylmetacrylate (PMMA), the material used as the samples for the laboratory experiments. We obtain the values for each parameter from the database of manufacturing companies that produce PMMA related products (Salem Ball Company, IDEMAT, Signal Processing), the material database from Massachusetts Institute of Technology and the Mechanical Engineering Department of the University of California San Diego.	24
Table 2.3. The brands and the technical characteristics of the glue we use in the preparation of sample 2 and the fracturing fluid that we use for the experiments.	27
Table 2.4. Descriptions for the viscometer and the seismic, pressure, and video equipment used in the experiments.....	29
Table 2.5. Acquisition rates for the seismic, pressure and camera data collected during the experiments.....	29
Table 3.1. Event classifications of the microseismic events identified in the two experiments. We classify the events based on the results of the two experiments using frequency, duration of the events, amplitude and whether the events are an individual event or a cluster of events.	58
Table 3.2. Variations in the in the location of the events based on the event type for experiment 2	61
Table 4.1. The scaling between the laboratory (L), magmatic (V) and hydraulic fracturing (HF) environments based on the inverse relationship between the source dimension and the dominant frequency (Aki and Richards 2002).	92

Table A.1 Fracture toughness, K , the Young's modulus, E , and the Poisson's ratio, ν , for polycarbonate and the upper and lower boundaries of shale, and the fluid viscosity, μ , and the flow rate, Q_0 of the fracturing fluid used to fracture these three different materials. The values listed in this table are from the personal communication with Dr. Juan Lorenzo and *Dr. Arash Dahi-Taleghani (Table 2.1 contains properties for PMMA)106

Table A.2. The parameters used in the calculations of non-dimensional toughness in order to determine the more suitable material to use as the samples in the experiments between PMMA and polycarbonate. The values listed in this table are from the personal communication with Dr. Juan Lorenzo and *Dr. Arash Dahi-Taleghani.....106

Abstract

There are uncertainties in the exact mechanisms that occur during hydraulic fracture growth and the interactions of hydraulic fractures with pre-existing fractures. In some cases fracture growth is seen to be purely tensile and in others a combination of tensile and shear mechanisms. In order to develop a seismic model to explain fracture propagation, there is a need for a complete classification of the microseismic events occurring during fracturing using waveform characteristics such as frequency, duration and magnitude. This classification would allow for more accurate prediction of the behavior of the hydraulic fracture from its initiation to when it intersects a natural fracture.

Due to the complicated nature of geological structures, it would be crucial to look at the microseismic events in the controlled laboratory environment to differentiate the physics of the problem from environmental factors. Laboratory experiments, however should be scaled correctly to mimic the real field-scale problem. Two experiments are conducted using polymethyl methacrylate (PMMA) samples; one with a pre-existing fracture and the other without this feature. Microseismic data is collected during each experiment from 8 stations with 3 sensors each, in a Galperin arrangement, on the samples. Pressure data and camera data are also collected to examine the changes in pressure and the growth of the fracture throughout the experiment. By spectral, qualitative and quantitative analysis of the data, we present a catalog of microseismic event types and propose several mechanisms for their differences. It is expected that these microseismic events show elements of both tension and shearing related to the opening and closing of fractures, fracture propagation and interaction with pre-existing cracks.

Chapter 1 Introduction

1.1 Problems and Objectives

Microseismic monitoring is an important tool in controlling the growth of hydraulic fractures created due to the over-pressurization of fluid within a rock (Economides and Nolte, 2000; van der Baan et al., 2013). Previous studies of the microseismic events generated during hydraulic fracturing have identified event types based on their frequency, amplitude and duration (Das and Zoback, 2011; Maxwell, 2011; Eaton et al., 2013; Maxwell et al., 2015). There remains lacking, a model describing the microseismic event types and the deformation causing their occurrence that can be applied across the field.

A classification of the types of microseismic events, based on the characteristics of frequency, amplitude and duration for each temporal stage of fracturing, ensures that fracture growth is better monitored and controlled. The locations of the events as they occur in time along with their classifications give a clearer understanding of how the microseismic event characteristics change at the initiation of the hydraulic fracture, during fracture propagation, when fluid flows within the fracture and during the interaction of the hydraulic fracture with a pre-existing natural fracture.

We conduct laboratory experiments using an isotropic material, as a simple case, to simulate hydraulic fracture formation. During the experiments, we collect seismic, pressure and video recording to identify expected types of microseismic events and their probable sources during fracture growth. The laboratory experiments allow for a controlled setting without the noise of the fracturing environment, and the time and economic constraints of the industry.

The main goal of this study is to categorize the microseismic events that occur in laboratory fracturing experiments based on qualitative descriptions as well as spectral analysis of the microseismic events. We examine the locations of each event at the time of their occurrence with respect to the growing fracture. Additionally, we identify unique features of the microseismic event characteristics, fracture dimensions and pressure responses that occur as a result of the interaction of the newly created fracture with a pre-existing sealed fracture.

1.2 Microseismicity associated with hydraulic fracturing

From previous work on microseismic monitoring during hydraulic fracture growth, three main event types are identified- high frequency events (Eaton et al., 2013), low frequency events (Eaton et al., 2013) , and low frequency tremor (Das and Zoback, 2011; Eaton et al., 2013; Tary et al., 2014).

High frequency events are microseismic events with a frequency greater than 100 Hz, with a duration of less than 5 seconds (Figure 1.1) (Eaton et al., 2013). There are two types of low frequency events with frequencies of less than 100 Hz. A single low frequency event has a duration of 20 seconds and less (Eaton et al., 2013). Another type of low frequency event called low frequency tremor (Eaton et al., 2013) or long-period, long-duration (LPLD) events (Das and Zoback, 2011) has a lower frequency range of 10-80 Hz lasting up to 100 seconds.

The high frequency events appear related to the brittle deformation during hydraulic fracture formation (Eaton et al., 2013). In oil and gas fields which contain complex fracture networks, more high frequency events may occur and thus more brittle deformation (Eaton et al., 2013).

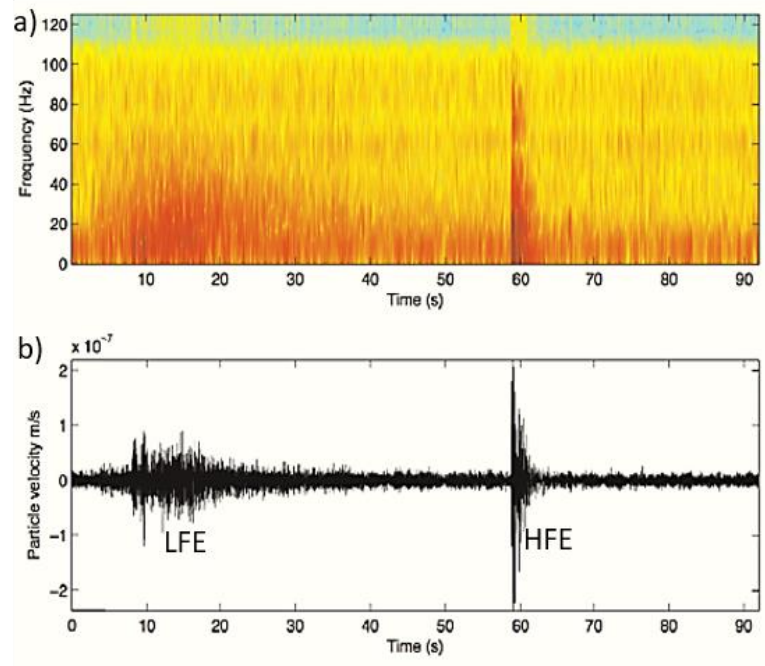


Figure 1.1. Low frequency and high frequency events from hydraulic fracturing treatment in the Montney Field in north-eastern Canada (Eaton et al., 2013). a) The frequency spectrum of the two events. The LFE corresponds to frequencies with a maximum of 60 Hz while HFE corresponds to maximum of ~ 110 Hz. b) The signal (measured in velocity) of the low frequency event followed by the high frequency event.

1.3 Seismicity related to magmatic fracture growth

The main events associated with magmatic fracture propagation are A-type, B-type, and volcanic tremor. A- and B-type events are volcano-tectonic earthquakes that generated due to the migration of magma through dykes and pathways within the volcano by shear or tensile fracturing (Zobin, 2012). Volcanic studies link A-type and B-type seismic events to the brittle deformation that occurs during the eruptive process in volcanoes.

A-type events have high frequencies consisting of frequencies > 5 Hz lasting for 15-30 seconds while B-type events have frequencies between 3 -5 Hz lasting 30 seconds (Wasserman, 2002). Although, A- and B-types have similar characteristics, one of the main

differences is in the waveforms themselves. P and S waves are clearly identifiable in the A-type while S waves in the B-type cannot be easily identified (Wasserman, 2002; Zobin, 2012).

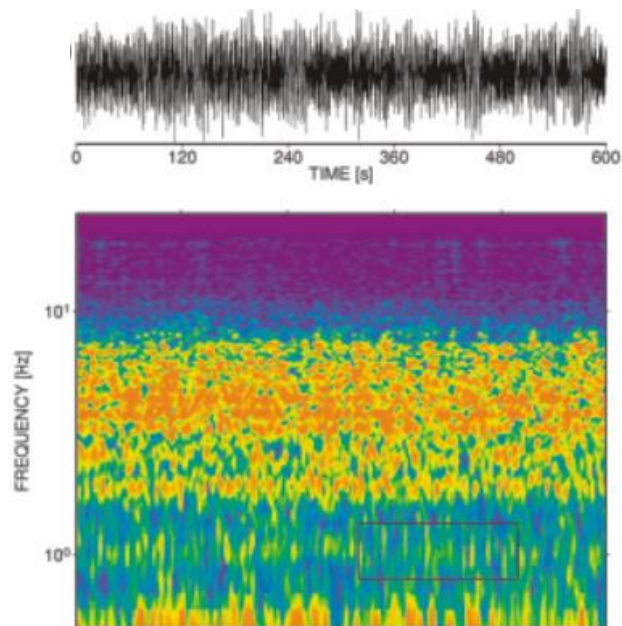


Figure 1.2. An example of volcanic tremor. These events are characterized by frequencies of less than 1 Hz for a duration lasting from a minute to months (Wasserman, 2002).

Volcanic tremor has a frequency < 1 Hz lasting from 60 seconds to several months long often occur in relation with each other (Zobin 2012). Low-frequency events, with frequencies between 1-3 Hz, occur closely in time and combine to form a long duration, long period signal (tremor) (Wasserman, 2002).

1.4 Comparison of the seismicity in hydraulic and magmatic fracturing

Magmatic and hydraulic fractures form in different environments and under different conditions such as temperature and scaling. For instance, different types of magma from rhyolitic to dacitic to andesitic to basaltic magmas melt within the range $\sim 600 - 1400$ °C respectively and thus, at these temperatures, the magma can migrate within the volcano to form dykes and sills (Zobin, 2012). In the case of hydraulic fracturing, the temperature range

depends on the temperature of the formation and the fracturing fluid used, but a typical range of temperatures observed in the field is ~200- 240 °C (Jones and Britt, 2009). The temperature at which the fracturing occurs, is important because temperature affects the viscosity of the fracturing fluid which can affect the rate of growth (Zobin, 2012) and which factors such the material toughness or fluid viscosity is dominant in controlling fracture growth (Detournay, 2004). The scales for magmatic fractures and hydraulic fractures vary with magmatic fractures reaching lengths of several kilometres (Zobin, 2012) and hydraulic fractures with average lengths of several hundred meters (Jones and Britt, 2009).

Although there are differences in the conditions of the formation of hydraulic and magmatic fractures, there are similarities related to the formation of both types of fractures. Hydraulic and magmatic fractures are both fluid-driven fractures. The initiation and growth of both types of fractures occur as a result of tension, shearing or a combination of the two mechanisms (Zobin, 2012; Maxwell, 2014)

From previous studies on hydraulic fracturing and magmatic fracturing in the field and in the laboratory, there are categories of seismic events that occur in both environments. Different deformation mechanisms produce specific types of seismic events. Three main categories of event types are common to the hydraulic and magmatic environments- high-frequency events, low-frequency events, and low frequency tremor. The frequency content of the microseismic event gives us an estimate of the rate and duration of the slip producing the event (Maxwell, 2014).

In both the hydraulic fracturing and magmatic fracturing environments, the high-frequency and low-frequency events have a shorter duration than the low frequency tremor. The high-frequency and low-frequency events in the hydraulic fracturing environment do not

exceed 20 seconds while the low frequency tremor (LPLD) can last up to 100 seconds. In the magmatic fracturing environment, high-frequency and low-frequency events last up to 5 seconds while the low-frequency volcanic tremor lasts for over 60 seconds (Table 1.1).

Table 1.1. Summary of the seismic events related to magmatic fracturing and hydraulic fracturing in the oil and gas industry grouped by their frequency types. (¹ Maxwell 2011; ²Eaton et al., 2013; ³Tary et al., 2014; ⁴ Wasserman, 2002; ⁵ Zobin, 2012)

Event	Magmatic fracturing ^{4,5}			Hydraulic fracturing ^{1,2,3}		
	Type	Frequency (Hz)	Duration (s)	Type	Frequency (Hz)	Duration (s)
High Frequency	A-type	>5	15- 30	High frequency events (HFE)	> 100	< 5
	B-type	~ 5	30			
Low frequency	B-type	< 5	30	Low frequency events (LFE)	< 100	< 20
Low frequency tremor	Volcanic tremor	< 1	> 60	LPLD and associated micro-earthquakes	10-80	10-100

Hydraulic fractures form by a dominantly tensile opening mechanism (Maxwell, 2014) which likely occurs aseismically. Aseismic deformation is deformation that occurs without detected seismicity (Maxwell, 2014). During hydraulic fracturing, most reservoir rocks are not likely to produce detectable microseismic events by tensile deformation (Maxwell, 2011). The rocks are not sufficiently strong in tension and do not produce enough seismic energy by tensile failure to create an observable seismic event (Maxwell, 2011). It is also possible that aseismic deformation can produce microseismic events with amplitudes too low to appear above the noise level and with frequencies too low to be measured by the seismic recording instruments (Maxwell, 2011). At the tip of the fluid-driven fracture and the region just behind the tip, the fracture opens and low frequency, low amplitude events may occur here. Fracture

opening is a slow process (Maxwell, 2011) and slow deformation may generate low frequency seismic events (Maxwell, 2011).

During hydraulic fracturing in the field, shear deformation occurs as a result of the failure of pre-existing fractures before and after the intersection of the hydraulic fracture (Maxwell, 2014). There are models in magmatic fracture propagation based on field and experimental data, where the stresses near the tip of the magmatic fracture are strong enough to favour shear failure along pre-existing faults far away and adjacent to the growing magmatic fracture (Zobin, 2012).

In general, when a fluid-driven fracture approaches a sealed pre-existing fracture, the tensile stresses at the tip of the growing fracture interact with the normal and shear stresses acting on the pre-existing fracture (Dahi Taleghani, 2009; Dahi-Taleghani and Olson, 2011). The influence of the stresses of the hydraulic fracture on the pre-existing fracture depends on the angle of approach and where along the length of the pre-existing fracture, the hydraulic fracture approaches. For instance, the hydraulic fracture can grow towards the center or the tips of the pre-existing fracture. From a study, modelling the interaction of a hydraulic fracture and a pre-existing fracture using the Material Point Method (Aimene and Nairn, 2014), we can observe how a differential stress field develops before the two fractures intersect (Figure 1.3). At the tips of the pre-existing fracture, we can observe a tensile stress develops. This can promote the opening and potential slip of the pre-existing fracture (Figure 1.3) (Aimene and Nairn, 2014).

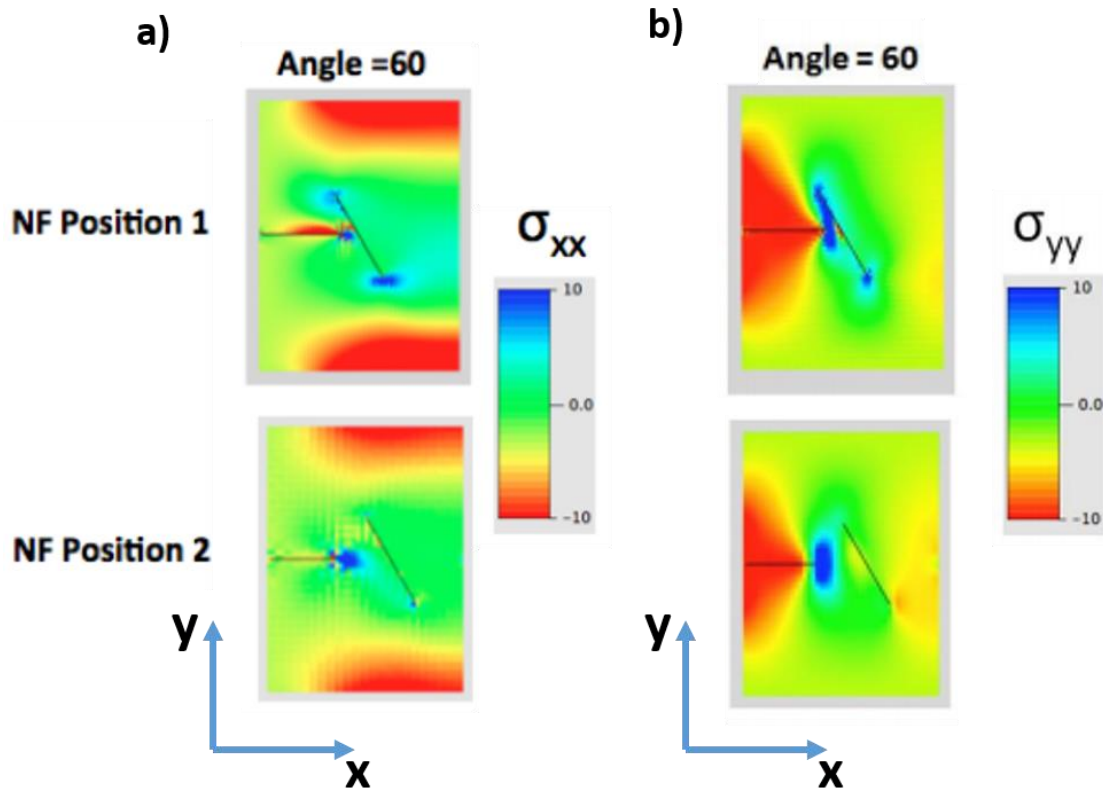


Figure 1.3. Material Point Method modelling for the variation in stresses as a hydraulic fracture approaches a pre-existing fracture at an angle of 60° , with a pressure of 10 MPa applied to the hydraulic fracture (Aimene and Nairn, 2014). The two horizontal stresses are equal i.e. anisotropy, measured as the ratio of the two horizontal stresses, is equal to 1. At NF position 1 the hydraulic fracture is closer to the pre-existing fracture than NF position 2. a) shows the variation of stress in the x direction while b) shows the variation of stress in the y direction. The blue color shows the tensile stresses and red shows compressional stresses.

If the stresses at the tip of the fluid-driven fracture are large enough to overcome the strength of the sealing material and the normal stress on the pre-existing fracture, debonding occurs even before the fractures intersect (Dahi Taleghani, 2009; Dahi-Taleghani and Olson, 2011). Debonding can cause slip along the pre-existing fracture (Aimene and Nairn, 2014; Dahi Taleghani and Olson, 2014). If the slip releases enough energy, a detectable microseismic event occurs (Aimene and Nairn, 2014; Dahi Taleghani and Olson, 2014). Extended finite modelling results (Dahi Taleghani, 2009) demonstrate the debonding of a growing fluid-driven fracture (blue) as it

approaches a pre-existing fracture (grey) at a non-orthogonal angle (Figure 1.4). The opened zone along the pre-existing fracture is asymmetric (Dahi Taleghani, 2009). The stress field of the hydraulic fracture “pulls” one section of the pre-existing fracture (Aimene and Nairn, 2014). At the same time, the stress field of the part of the pre-existing fracture not affected by the hydraulic fracture, produces a compressional stress acting to close the hydraulic fracture (Aimene and Nairn, 2014).

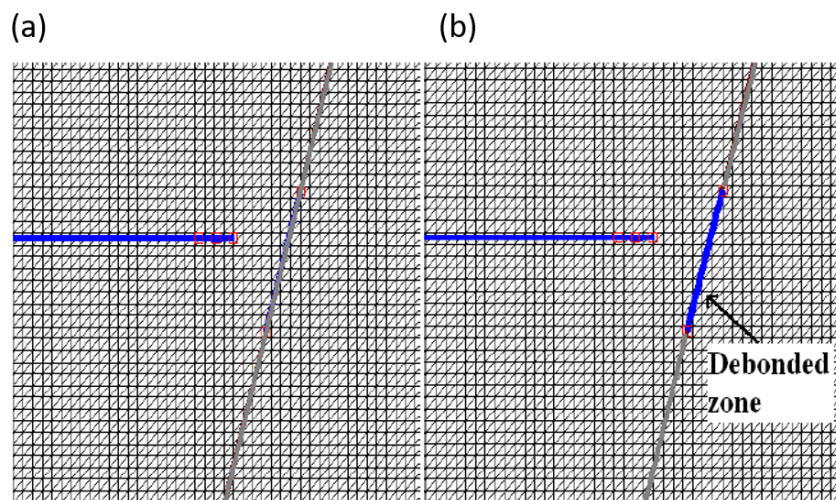


Figure 1.4. Extended finite element modelling results of the debonding that takes place as a fluid-driven fracture approaches a non-orthogonal pre-existing sealed fracture (Dahi Taleghani, 2009) (a) A fluid-driven fracture (blue) approaches a pre-existing sealed fracture (grey). (b) The debonded zone (blue along the pre-existing fracture (grey)) is asymmetric and thus can result in the formation of a one direction deflection along the pre-existing fracture when the two fractures intersect.

Although previous studies do not explicitly link high-frequency shear microseismic events occurring in the field (Maxwell, 2014) directly to a specific mechanism for their generation, the shear slip of the pre-existing fracture is a likely cause of the generation of high-frequency seismic events. In magmatic environments, the shear slip along pre-existing fractures are also the probable cause of higher frequency volcano-tectonic earthquakes, A-type and B-type (Zobin, 2012).

The intersection of the fluid-driven fracture and the pre-existing fracture can result in three scenarios: arrest, crossing or deflection of the fluid-driven fracture (Warpinski and Teufel, 1987). We consider when the fluid-driven fracture deflects into the pre-existing fracture, that it intersects based on our laboratory fracturing experiment involving a sample containing a pre-existing sealed fracture. In this case, the fluid-driven fracture approaches the pre-existing fracture at a non-orthogonal angle and asymmetric debonding of the pre-existing fracture may occur (Dahi Taleghani, 2009).

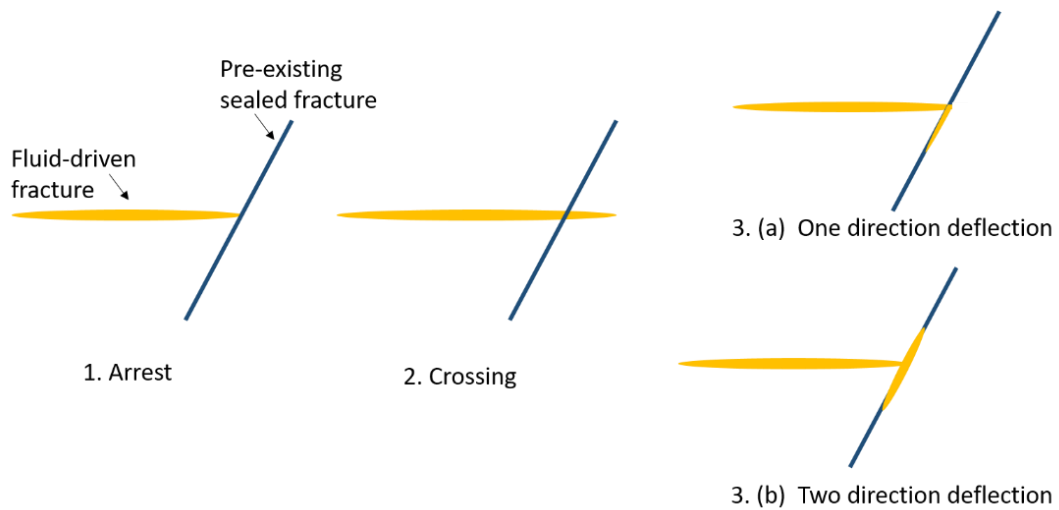


Figure 1.5. The possible scenarios when a fluid-driven fracture intersects a natural fracture (Warpinski and Teufel, 1987; Dahi Taleghani, 2009).

The asymmetry of the opening causes the growing fracture to propagate in one direction along the pre-existing fracture upon their intersection (Aimene and Nairn, 2014; Dahi Taleghani and Olson, 2014). The fluid-driven fracture seeks the easiest path of propagation (Dahi Taleghani and Olson, 2014). The energy release for the diversion of the growing fracture into the pre-existing fracture is greatest compared to the other two possibilities of arrest or crossing (Dahi Taleghani and Olson, 2014). As the fluid from the fluid-driven fracture flows into the pre-existing fracture, dilation of the fracture is likely to occur at

the intersection and along the pre-existing fracture in order to accommodate the new volume of fluid (Dahi Taleghani and Olson, 2014).

In studies on hydraulic fractures in the field (Maxwell et al., 2015; Rutledge et al., 2015), shear deformation also occurs when the tip of the fluid-driven fracture encounters bedding planes and induces slip along the bedding planes. The shear failure which involves the brittle deformation of the rock, may produce high-frequency shear microseismic events that occur in the field (Maxwell, 2014).

Low-frequency events and tremor have different possible mechanisms for their generation based on the environment and the location of their source. During fluid-driven fracture propagation, the pressure of the fluid fluctuates as the fluid moves into and through the fracture. The fluctuation occurs as a result of different trigger mechanisms such as in magmatic settings where the release of energy from the shear deformation of the fracture as it propagates, remains within the fluid of the fracture (Neuberg et al., 2006). Another possible cause of the fluctuations is the expansion and collapse of the fracture as fluid fills the void created by the growing fracture (Chouet, 1986; Wassermann, 2002; Zobin, 2012). This expansion and collapse of the fracture is the resonance of the fracture and can result in the occurrence of clusters of low-frequency earthquakes and low-frequency tremor (Chouet, 1986; Wassermann, 2002; Zobin, 2012; Tary et al., 2014).

Another possible source of the clusters of low frequency events and low-frequency tremor observed in the field during fluid-driven fracturing is shear deformation along faults and pre-existing fractures (Shelly et al., 2007). In non-volcanic natural seismicity, the driving mechanism of these type of events along major faults and plate boundaries is slow slip along the faults (Shelly et al., 2007). The low-frequency tremor, LPLD events, observed in the field

during hydraulic fracturing may originate from the resonance of the fracture or slow slip along pre-existing fractures. In a study on the LPLD events in the Barnett Shale, although resonance is not completely ruled out as a source of the events, slow-slip along a pre-existing fault is the dominant mechanism producing the events, because the events occur consistently in the same orientation and direction of the natural fractures and pre-existing faults (Das and Zoback, 2013). The location of the source of the LPLD events can help distinguish the mechanisms producing the events.

In preparation for hydraulic fracturing in the field, the drilling of the well can introduce flaws as tiny fractures close to the wellbore (Frash, 2007). Drilling can also induce heterogeneous stresses around the wellbore. The variation in the stresses around the wellbore can lead to wellbore breakout, the enlargement of the cross section of the wellbore (Grandi et al., 2002) and differential loading, where the differences between the vertical stress and the stresses at different points around the wellbore are not the same (Frash, 2007; Wu et al., 2007). If the pre-crack is not perfectly in the plane of the direction of the maximum principal direction, the fluid-driven fracture twists or turns from the original direction of the pre-crack to an orientation that is parallel to the maximum principal stress (Frash, 2007). The fracture preferentially grows parallel to the maximum principal stress as this direction requires the least energy for propagation.

Differential loading and the twisting that occurs as the fracture grows from an asymmetric pre-crack can cause the formation of mixed mode III shearing along the fracture surface (Wu et al., 2007). Segmentation of the fracture can result from mode III shearing where sections of the fluid-driven fracture may be offset from main plane of the growing fracture (Frash, 2007; Wu et al., 2007). High frequency shear microseismic events that occur

in the field during hydraulic fracturing (Maxwell, 2014), may occur close to the wellbore as a result of the formation of these mode III (shear) fractures.

Based on the probable mechanisms that produce each type of microseismic event, we can group the events by the likely locations of their sources with respect to the growing fracture or pre-existing fractures that may be present. We can use this grouping as a guide for the expected types of seismic events we may observe throughout our laboratory experiments.

Table 1.2. A proposed model showing the probable origins and locations of the seismic events occurring in association with fluid-driven fracture growth. The origins and locations are likely based on the assumptions that particular mechanisms at different stages of the fracture growth will produce seismic events of different characteristics such as frequency or amplitude (¹ Aimene and Nairn, 2014; ² Chouet, 1986; ³ Dahi Taleghani and Olson, 2014; ⁴ Eaton et al, 2013; ⁵ Frash, 2007; ⁶ Maxwell, 2014; ⁷ Maxwell et al., 2015, ⁸ Rutledge et al., 2015; ⁹ Shelly et al., 2007; ¹⁰ Tary et al., 2014; ¹¹ Wasserman, 2002; ¹² Wu et. al, 2007; ¹³ Zobin, 2012).

Event type	Probable origin of events	Probable Location
High frequency events	^{1,3,4,6,7,8} Shear slip as tip intersects pre-existing fractures or weaknesses	At tip/edge of the fracture
	^{5,12} Mode III shearing due to twisting of the fracture and differential loading	Near the wellbore (during hydraulic fracture growth in the field
	^{1,3,13} Stress field interaction	Along pre-existing fracture
Low frequency event and Low frequency tremor ^{1,2}	⁶ Slow aseismic opening	At tip/edge of the fracture
	^{2,4,10,11,13} Fluid movement; resonance	Within the body of the fracture
	^{1,3,9} Slow shear slip	On pre-existing fracture/fault

1.5 Key Concepts

1.5.1 Fluid-driven fractures

A fracture is a discontinuity surface in a solid (van der Pluijm and Marshak, 2004). Fluid-driven fractures are fractures that are generated wholly or partly due to the overpressure of an internal fluid within the rock and can be generated naturally or through man-made activities (Brenner and Gudmundsson, 2004). Natural fluids that can generate fluid-driven fractures include groundwater (through hydrothermal veins), magma, oil, gas and in the case of hydraulic fracturing of unconventional petroleum reservoirs, fracturing fluid (Brenner and Gudmundsson, 2004). In some cases, the fluid generating the fractures may mineralize to fill dykes or veins or the fractures remain open as joints (Brenner and Gudmundsson, 2004).

1.5.2 Fluid-driven fracture mechanics

The three principal stresses - σ_1 , the maximum principal stress, σ_2 , the intermediate principal stress, and σ_3 , the minimum principal stress, control fracture propagation and geometry (Jones and Britt, 2009). The fracture will initiate when the fluid-pressure is greater than the closure stress (σ_c) that is the stress that keeps the fracture closed- σ_1 for horizontal fractures and σ_2 for vertical fractures (Jones and Britt, 2009). The fractures propagate in the direction perpendicular to the minimum principal stress (σ_3) (Jones and Britt, 2009).

Fracture propagation, the continued growth of the fracture after the initiation, depends on the stress intensity factor,

$$K_I = \Delta\sigma_I \sqrt{\pi * L}, \quad 1.1$$

where K_I is magnitude of the stress at the fracture's tip (stress intensity factor), $\Delta\sigma_I$ is the driving stress, and L is the fracture length (Liu, 1996). The driving stress relates to the fluid pressure in the fractures as,

$$\Delta\sigma_I = P_f - \sigma_n,$$

1.2

where P_f is the fluid pressure in fracture and σ_n is the normal stress on the entire sample (Liu, 1996). When K_I is equal to K_{IC} (the fracture toughness or strength of the material), propagation occurs (Dahi-Taleghani and Olson, 2011).

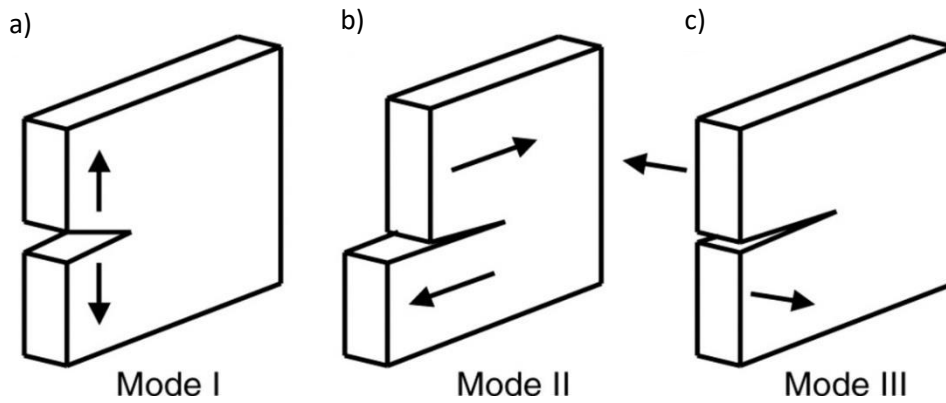


Figure 1.6. Block diagrams showing the three modes of crack surface displacements. Mode I (a) illustrates a tensile-mode crack displacement which is the equivalent of opening, Mode II (b) illustrates a shear-mode crack displacement which is in plane shearing, and Mode III (c) illustrates a shear-mode crack displacement which is out-of-plane shearing. (van der Pluijm and Marshak 2004).

Depending on the differential stress ($\sigma_1 - \sigma_3$) and the rock tensile strength, the type of failure that can occur varies (Sibson, 2000). For brittle fractures, the three major deformational modes are purely extensional (tensile), extensional-shear hybrid and purely shear (Sibson, 1996). A Mohr-diagram (Figure 1.6) describes the stress variations and the criteria for failure (Maxwell, 2014) for a rock with tensile strength, T , with a friction coefficient, μ_i , given by the slope of the line labelled μ_i on the graph (Sibson, 1996). The diameter of the semi-circle shows the differential stress ($\sigma_1 - \sigma_3$). Mode I, or tensile failure, occurs when there is a negative tensile stress and the differential stress is relatively small (Figure 1.6 (i)) (Maxwell, 2014). Mode II and III, shear fracturing occurs when the Mohr semi-circle intersects the failure curve (Figure 1.6 (iii)) (Maxwell, 2014). Mode II and III fractures

typically form at an angle $2\theta_s$ typically in the range of $\pm 30^\circ$ from the direction of the maximum principal stress (Sibson, 1996). Extensional-shear fracturing which is a mixture of tensile and shear failure occurs where the tensile stress is negative and where the shear stress is large enough to cause failure (Figure 1.6 (ii)) (Maxwell, 2014).

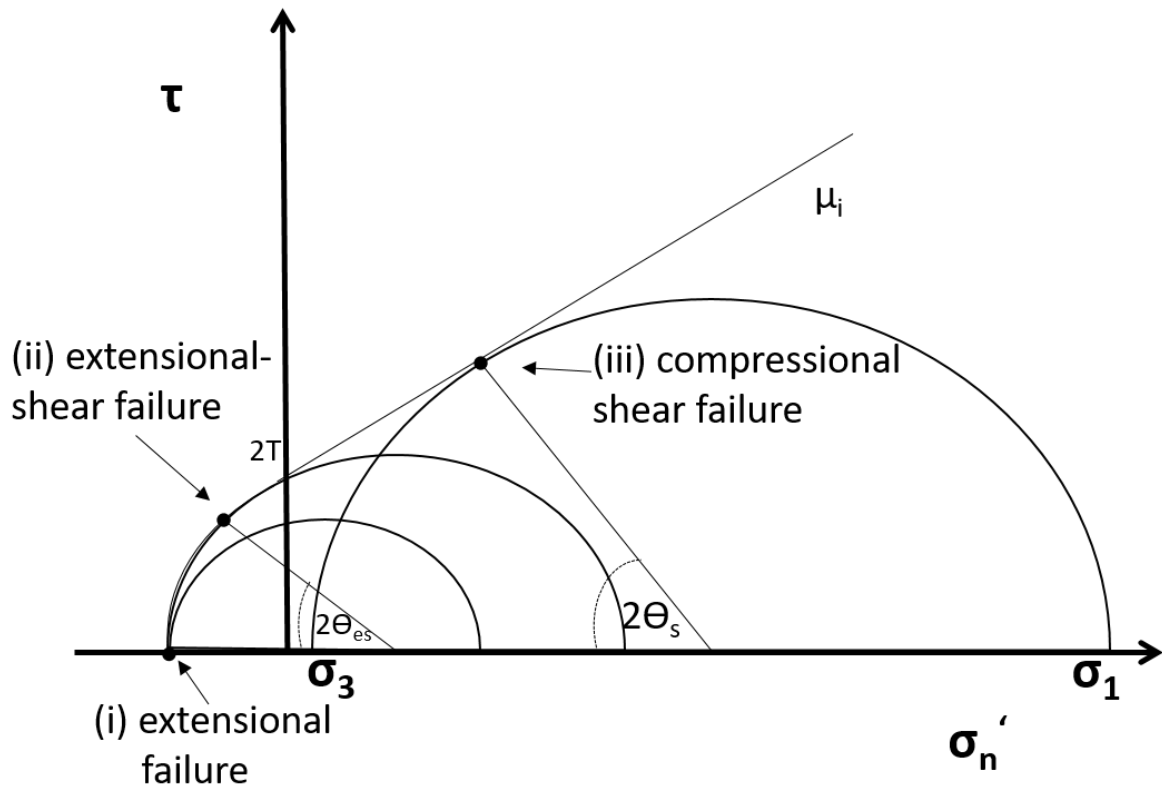


Figure 1.7. Composite Griffith-Coulomb failure envelope shown on a Mohr diagram for an intact, homogenous, isotropic rock with a tensile strength, T . The Mohr diagram shows shear stress, τ , versus effective normal stress ($\sigma_n - P_f$, where σ_n is the normal stress and P_f , fluid-pressure in fracture). σ_1 is the maximum principal stress, σ_3 is the minimum principal stress, μ_i is the coefficient of friction along a fault and θ_s is the angle between the shear fracture and σ_1 . The criteria for different failure modes (i) extensional (mode I), (ii) extensional-shear, and (iii) compressional shear failure for a rock are shown (mode II and mode III) (adapted from Sibson, 2000).

For pre-existing fracture or fault reactivation, using the Coloumb criterion, slip will occur when,

$$\tau \geq S_o + \mu_i \sigma_{n\theta_s}, \quad 1.2$$

where τ is the shear stress on the fault plane, S_o is the cohesion of the interface, μ_i is the friction coefficient of the fault plane and $\sigma_{n\theta_s}$ is the normal and shear stress acting on the interface with an angle θ_s to the maximum principal stress (Sibson, 1977).

1.6 Application of the proposed seismic model

The proposed model seeks to generalize the mechanics involved in fluid-driven fracture propagation to be applied in different environments such as volcanoes where magmatic fractures form, hydraulic fracturing stimulations in the oil and gas industry and our laboratory study of hydraulic fracturing (Figure 1.3). By characterizing the types of microseismic events and deformation expected for each phase of fracture growth, operators may be better able to predict the behaviour of the fractures. For instance, we can consider an area undergoing hydraulic fracturing treatment where low-frequency tremor and clusters of low-frequency seismic events occur consistently. If these events occur in a location that is not directly interacting with the hydraulic fractures growing, then operators may be more confident in linking the occurrence of the low-frequency tremors and events with slow slip along a fault. The operators can then interpret the occurrence of these events as the reactivation of a pre-existing fault and can stop or adjust the treatment to reduce the risk of causing a major earthquake in that area. This has the direct impact of reducing the risk of induced seismicity that can occur when the hydraulic fractures cause pre-existing faults to slip (Dahi Taleghani and Lorenzo, 2011).

Additionally, the proposed seismic model can aid in the monitoring of the growth of magmatic fractures. Magmatic fractures aid in the transport of magma in the volcanic conduit and thus understanding the fluid and mechanical processes that occur to produce specific

seismic events improves our understanding of how volcanoes erupt. If volcanologists can match each type of earthquake with an associated mechanism confidently, then when they locate the earthquakes occurring, they will be better able to understand where the ascending magma is within the volcano and the type of deformation that may be occurring within the volcano at the time that the seismic event occurs. This in turn may help to improve predictions of volcanic eruptions.

1.6.1 Hydraulic fracturing process

Hydraulic fracturing is the breaking or brittle failure of a rock creating more permeability to allow for the movement of fluids out of the rock (Maxwell, 2011; van der Baan et al., 2013). This failure is usually accompanied by microseismic events which are earthquakes of a moment magnitude, $M_w < 0$ (Maxwell, 2014).

We use the process of the hydraulic fracturing of unconventional reservoirs as a basis for the development of the laboratory procedure to create the fluid-driven fractures. In the oil industry, hydraulic fracturing is carried out within tight gas reservoirs that have low permeability and porosity (van der Baan et al., 2013).

The process of hydraulic fracturing in the oil and gas industry, involves the injection of fluids containing proppants (solid particles of a specific size) under high pressure into the reservoir. Fractures form as a result of the injection of the fluid and the interconnectivity of fractures increases enhancing the ability of the reservoir to drain of hydrocarbons (van der Baan et al., 2013).

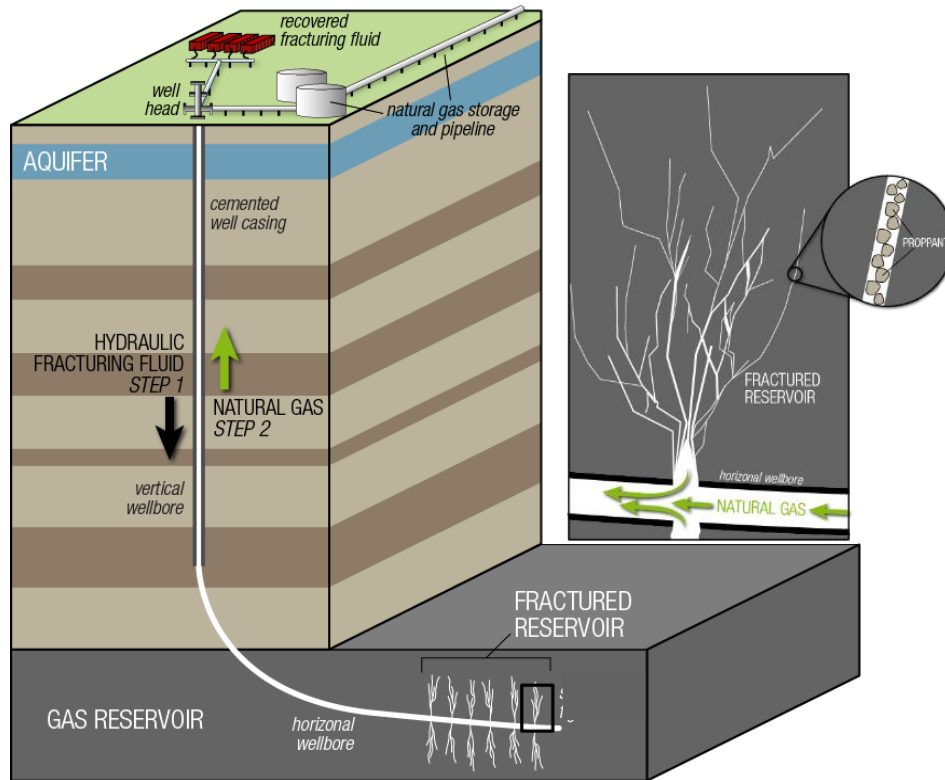


Figure 1.8. A schematic showing the steps in hydraulic fracturing. Hydraulic fracturing fluid is pumped into the target formation via deviated well (the well is initially vertical but is deviated so that it is horizontal within the target formation). The high fluid pressure results in the formation of hydraulic fractures which are controlled by maintaining a specific treatment pressure and monitoring the growth through microseismic monitoring (Zuppann and Steinmetz, 2014).

1.6.2 Microseismic monitoring during hydraulic fracturing in the industry

Microseismic monitoring is the passive monitoring of the microseismic events that are generated from the formation of hydraulic fractures. A microseismic event occurs if enough energy releases as geomechanical stress is released (Maxwell, 2011; 2014). The microseismic events not only provide us with information of the location of deformation, but in turn can tell us the orientation, dimensions, strength and mechanism of the deformation that occurs (Maxwell, 2014; Downie et al., 2010). The amplitude of the seismic event gives us an idea of the amount of energy released during the deformation. The frequency content gives us an estimate of the rate and duration of the slip or fracturing and the seismic wave radiation

pattern provides us with information on the source mechanism producing the event (Maxwell, 2014).

The main drawback of microseismic monitoring in the field is that the quality of the data determines the reliability of the results (Maxwell et al., 2010). In the field, unwanted signal, commonly called noise, generates from the hydraulic fracturing equipment such as logging tools, and other activities on the surface such the movement of vehicles like trains (Warpinski, 2009). The greater the amount of unwanted signal corrupting the seismic data, the lower the quality of the data as measured by the signal-to-noise ratio. The signal-to-noise ratio (SNR) is a measure of the event amplitude as compared to the noise amplitude and the higher the SNR of a seismic event, the better the recording of the seismic event (Warpinski et al., 1998). Low signal-to-noise ratios increase the uncertainty in the picking of microseismic events, analysing the waveforms (for frequency), and in locating the events (Maxwell et al., 2010). In the interpretation of the microseismic events from the field, the assumption is that the events that recorded during the treatment directly relate to the newly created fluid-driven fracture and stress changes around that fracture (Maxwell et al., 2010). However, there are instances where pre-existing faults are the source of the seismic events and can influence the growth of the hydraulic fractures (Maxwell, 2011) and thus where the microseismic events occur (Maxwell et al., 2010; Warpinski, 2013).

Chapter 2 Methods

2.1 Experimental Set-up

To test the seismic model for the initiation and propagation of hydraulic fractures, we conduct two laboratory experiments using samples made from polymethyl methacrylate (PMMA) (Table 2.1). Sample 1 is a PMMA block without a model fault (pre-existing sealed fracture) and sample 2 contains a model fault at an angle of 60° with an error of $\pm 0.083^\circ$ (Table 2.2). We use the samples with and without the fault to compare any differences we observe in the growth of the fracture and how these differences are reflected in the data that we collect.

Table 2.1. The dimensions of sample 1 and sample 2 used in the experiments (length, width and height of each sample). The error in measurements is ± 0.0005 m

Sample #	Dimensions (m)		
	Length	Width	Height
1	0.151	0.149	0.097
2	0.30	0.15	0.077

2.2 Laboratory Scaling

We need to consider the mechanisms involved in fluid-driven fracturing in the field in order to scale our experiments in the laboratory so that we can apply our laboratory results to what occurs in the field. Fluid-driven fracture growth occurs in either the viscous-dominated regime or the toughness-dominated regime (Detournay, 2004). We can use an analytical approach to determine which regime is dominated for fluid-driven fracturing which accounts for mechanisms involved in fluid-driven fracturing such as the deformation of the

rock, creation of new fractures, the viscous fluid flow within the fracture and the leak-off of the fracturing fluid into the permeable surrounding rock (Detournay et al., 2007). The material properties that represent these mechanisms are Young's modulus, E , Poisson's ratio, ν , the fracturing fluid's viscosity, μ , the rock's toughness, K_{Ic} , and the leak-off coefficient, C_l , for permeable rocks (Detournay et al., 2007). We can adjust our experimental materials and conditions to ensure that we conduct the experiment in the appropriate regime based on the analytical solution.

We use the dimensionless toughness, \tilde{K} , to distinguish between the two regimes, where

$$\tilde{K} = \frac{K'}{(E'^3 \mu' Q'_0)^{\frac{1}{4}}}, \quad 2.1$$

where K' , E' , and μ' are dimensionless rock toughness, Young's modulus and fluid viscosity, and Q'_0 is the pump rate,

defined as,

$$\mu' = 12\mu, \quad 2.2$$

$$E' = \frac{E}{1 - \nu^2}, \quad 2.3$$

and

$$K' = 4\left(\frac{2}{\pi}\right)^{\frac{1}{2}} K_{Ic} \text{ (Detournay, 2004)}. \quad 2.4$$

When $\tilde{K} \leq 1$, the regime is viscous-dominated where most of the energy dissipates in the fluid and the fracturing is independent of the toughness of the material (Detournay, 2004). When $\tilde{K} \geq 4$, the regime is toughness-dominated where most of the energy dissipates at the fracture tip and goes into the creation of the new fracture surface. The

toughness-dominate regime is greatly dependent on the toughness of the material (Detournay, 2004).

Magmatic fracture propagation varies widely from viscous to toughness dominated regimes dependent on the rock toughness and magma viscosity. Most sills (horizontal magmatic fractures) propagate in the viscous-dominated regime and in the transition between viscous- and toughness-dominated propagation along bedding planes. Bedding planes are lines of weaknesses within the rock so most of the energy for the fracturing goes to the fluid since the fracture does not need as much energy to create a new path (Bunger, 2008; Maimon et al., 2012). On the other hand, dykes (vertical magmatic fractures) have a wider range of viscous to toughness-dominated regimes because the formation of dykes requires more energy to create new fracture surfaces as dykes cut across bedding planes and propagate in rocks with varying toughness (Bunger, 2008; Maimon et al., 2012). Hydraulic fracturing treatments in the field are conducted in the viscous-dominated regime as shown by analytical, numerical and experimental studies (Garagash and Detournay, 2002; Detournay et al., 2007).

We conduct the experiment in the viscous-dominated regime since this regime is common to some types of magmatic fracturing and to hydraulic fracturing the field. To ensure that the experiment is conducted in the viscous-dominated regime, we choose polymethyl methacrylate (PMMA) for our samples and a fracturing fluid with a viscosity of 96,000 cps giving us a dimensionless toughness factor of approximately one indicating that the experiment allows for fracturing to occur in the viscous-dominated regime. If compared to the dimensionless toughness of the fracturing of shale in the field, we observe that it falls at

the upper limit of shale formations and thus we expect the PMMA to have similar deformational behavior to shale when fracturing the PMMA in the laboratory (Figure 2.2).

Table 2.2. Properties of polymethylmetacrylate (PMMA), the material used as the samples for the laboratory experiments. We obtain the values for each parameter from the database of manufacturing companies that produce PMMA related products (Salem Ball Company, IDEMAT, Signal Processing), the material database from Massachusetts Institute of Technology and the Mechanical Engineering Department of the University of California San Diego.

Property	Value	References
Density, ρ	1.19 g/cm ³	(Salem, 2014)
Poisson's ratio, ν	0.35	(M.I.T., 2014)
Tensile strength, T	48-76 MPa	(M.I.T., 2014)
Shear Modulus, μ	1700 MPa	(IDEMAT, 2003)
Young's Modulus, E	1800 – 3100 MPa	(IDEMAT, 2003)
Fracture Toughness, K_{IC}	0.8 – 1.75 MPa \sqrt{m}	(MAELABS, 2011)
P-wave velocity, V_p	2750 m/s	(Signal Processing, 2014)
S-wave velocity, V_s	1375 m/s	(Estimated from P-wave velocity)

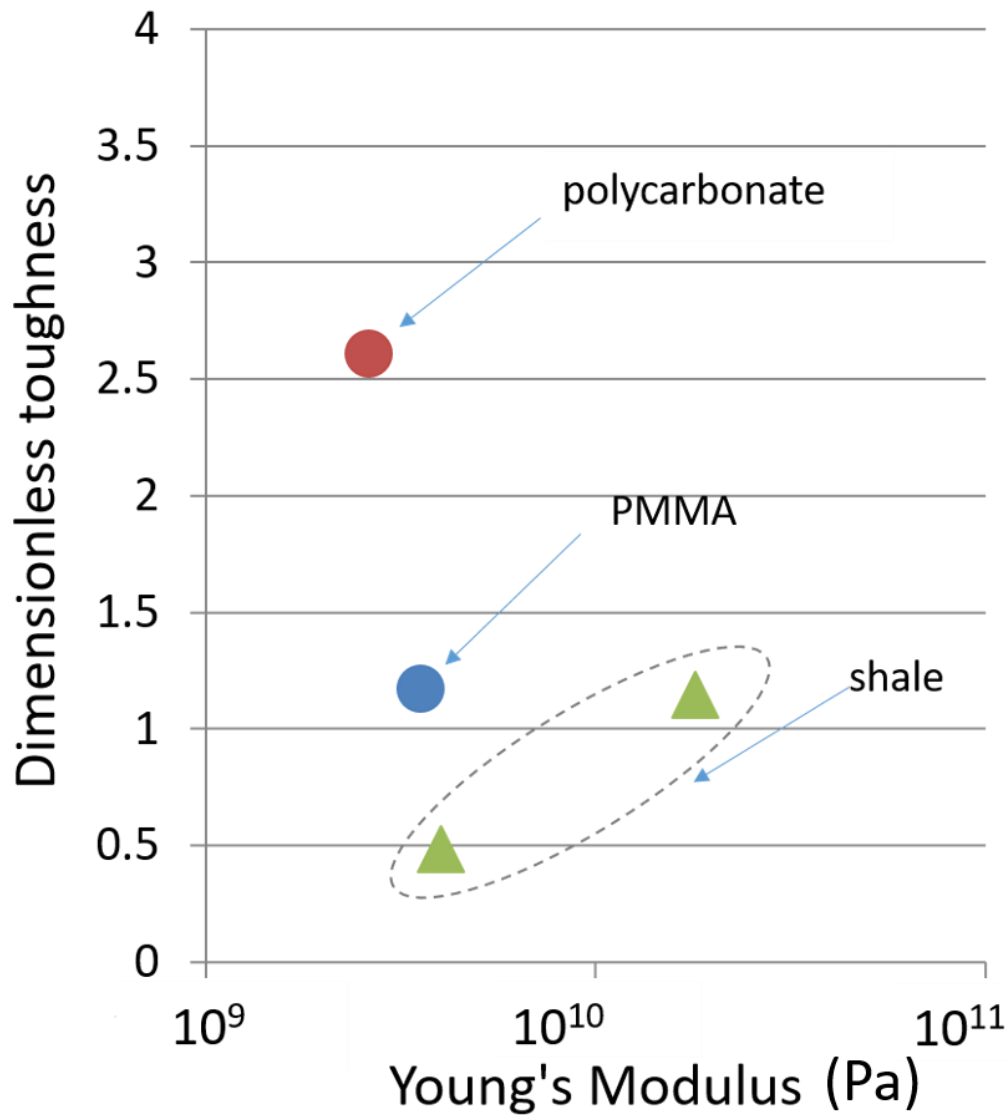


Figure 2.1. Dimensionless toughness for three materials- polycarbonate, polymethyl methacrylate (PMMA) and shale. We calculate the dimensionless toughness using equation 2.1 and inserting the appropriate values for each material and the parameters for the experiment (Appendix). PMMA is the material we choose to use as our samples because the experiment using PMMA and the conditions we set for the laboratory has a dimensionless toughness of approximately 1 which means that we expect the PMMA to deform similarly to shale in the hydraulic fracturing laboratory experiments. Additionally, PMMA is cheap, readily available and transparent. Since it is an isotropic, homogenous medium, we use PMMA as a simple case for the generation of our model.

2.3 Experimental Procedure

As part of the sample preparation, we polish the blocks of PMMA to enhance their transparency because after levelling the samples to an accuracy of ± 0.001 inches, the blocks are left dull and opaque (Sample Preparation, Appendix A).

After polishing the samples, we carry out steps similar to the hydraulic fracturing treatments in the field. We drill a borehole with a diameter of 0.0191 m to a depth of half of the sample's height (Table 2.2), in the center of each sample to act as a model well. In the field, perforations, which are small cracks created before pumping begins, provide entry points for the fracturing fluid to flow from the well into the rock helping to initiate fracture growth and propagation. For each sample, we create a horizontal pre-crack at the wellbore approximately 0.02 m in diameter with a thickness < 0.01 m. In the case of this study, σ_1 is in the horizontal direction and σ_3 is in the vertical direction, therefore the fractures created in the experiments are horizontal. After creating the borehole, unwanted stresses are introduced into the samples and to remove or reduce these stresses, we treat the samples with heat for thermal annealing (Sample preparation, Appendix A). To prevent air bubbles within the borehole when we begin the experiment, we fill the well with fracturing fluid before pumping begins.

We use a biaxial press to exert 1000 psi of pressure in two horizontal directions so that our created fracture grows horizontally and parallel to the maximum principal stresses. Our seismic sensors are placed on the top and bottom faces of our samples leaving these faces free from the application of any external stresses (Figure 2.1, Figure 2.2). We pump fracturing fluid into the sample at a constant flow rate of 4 $\mu\text{l}/\text{min}$ and we record seismic, camera and pressure data from the time pumping begins (Appendix).

Table 2.3. The brands and the technical characteristics of the glue we use in the preparation of sample 2 and the fracturing fluid that we use for the experiments.

Materials	Specifications
Glue	J-B Weld Product 8265-S; tensile strength 3960 psi applied to hold two separate pieces of sample with pre-existing fracture
Fracturing fluid	Mixture of glucose, sucrose and Kraft Kool-Aid (powdered mix drink containing sugar, citric acid, Vitamin C, Vitamin E, calcium phosphate, sweeteners, and artificial color (Kraft, 2015)); viscosity of 96,000 cps and a ratio of sucrose to glucose of 80:20 (Fracturing Fluid Preparation, Appendix B)

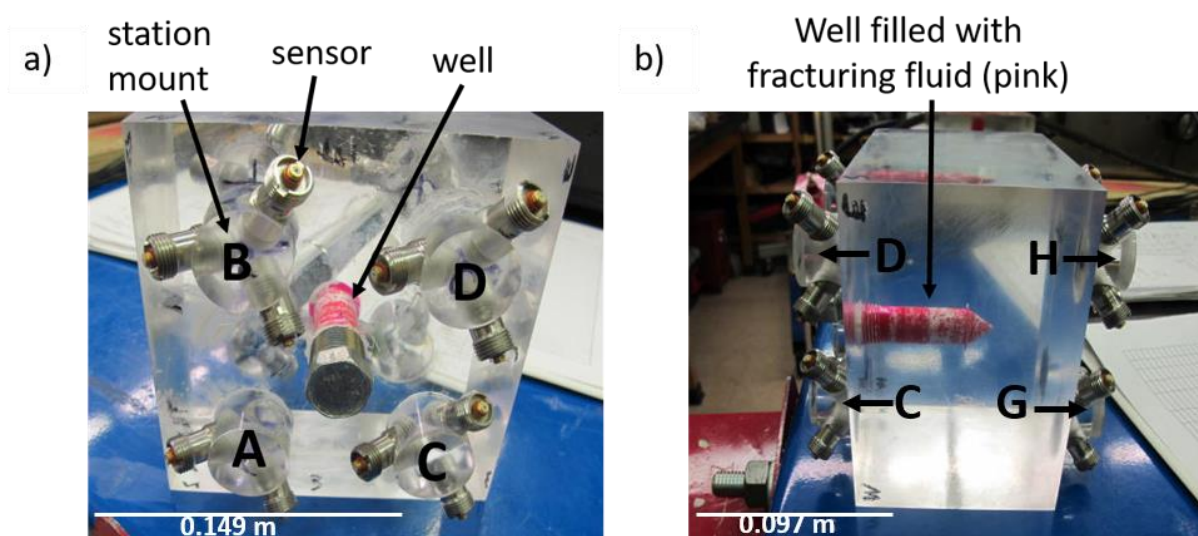


Figure 2.2. Sample 1 (without the fault) in top view (a) and side view (b) showing the stations on top of the sample labelled A-D (the four stations on the top of sample 1) with the three piezo-electric sensors attached to each station in a Galperin arrangement. Each sensor is connected to a channel that is part of the seismic acquisition system. The well, of diameter $\frac{3}{4}$ " is in the center of the sample (the pink fluid is the fracturing fluid used to fill the well before the experiment begins).

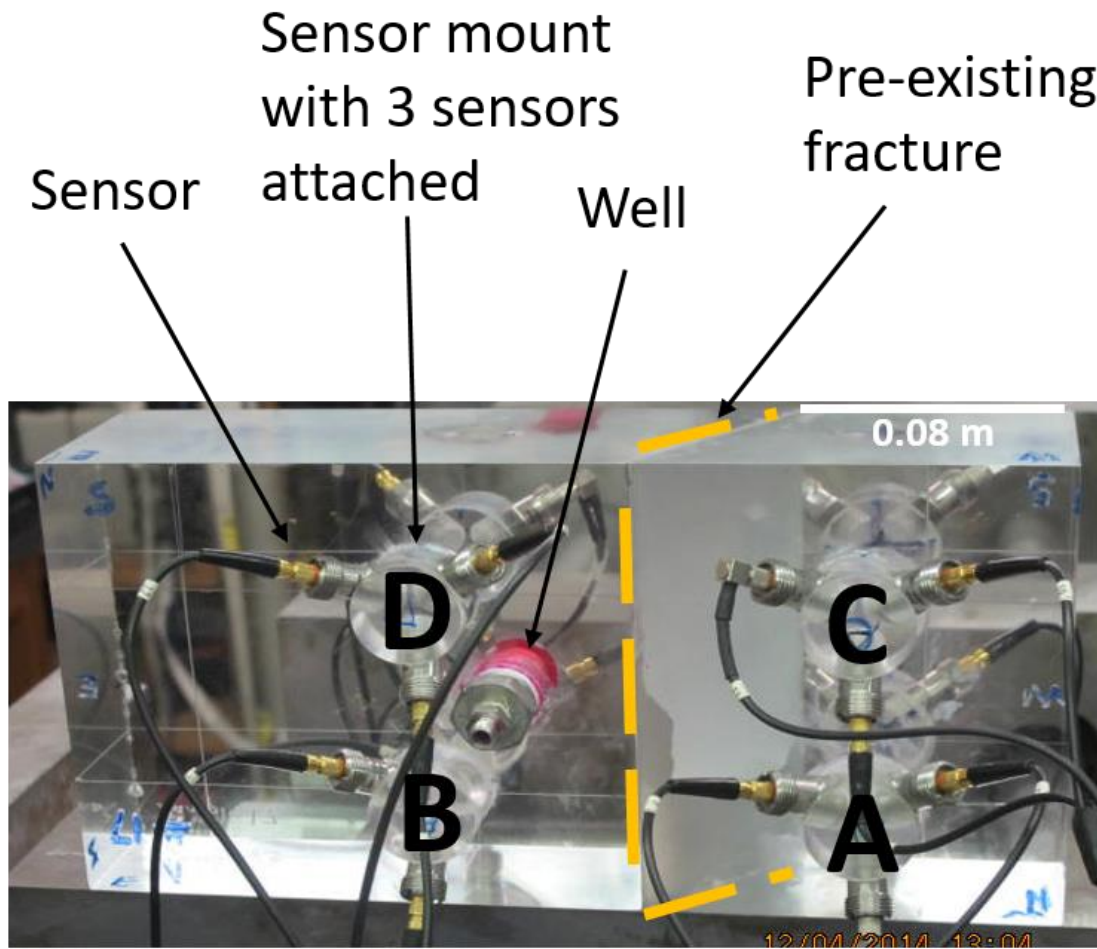


Figure 2.3. Sample 2 (with the pre-existing fracture) in top view. Sample 2 consists of 2 PMMA blocks each with one side pre-cut to fit together at an angle 60° . The two blocks are held together by JB-Weld (glue). The yellow dashed line represents the pre-existing fracture within sample 2. The stations on the top of the sample are labelled A-D with the three piezo-electric sensors attached to each station in a Galperin arrangement (Galperin, 1955). Each sensor is connected to a channel that is part of the seismic acquisition system. The well, of diameter 0.019 m is also filled with the fracturing fluid before the start of the experiment.

Table 2.4. Descriptions for the viscometer and the seismic, pressure, and video equipment used in the experiments.

Equipment	Specifications
Viscometer	Digital Viscometer (from Brookfield); 1% accuracy; 2% repeatability
Geophone	24 Piezo-electric sensors (KRNBB-PC Point contact sensor from ‘Steve Co.’); spectral response of 100kHz to 2.5MHz
Channels	2 boards of 12 channels (channels are devices to carry data from the sensor to the recorder (Schlumberger, 2016)) (National Institute of Standards and Technology (NIST) 24 from Ario Labs, LLC); channels are attached to each sensor
Syringe Pump	D-Series Syringe Pump and controller Model 100DM (from Teledyne ISCO); fracturing fluid is pumped into the sample using this pump
Wellhead Pressure meter	Industrial Pressure Transducer (Model 522 from Setra); at wellhead
Biaxial Press	Two pistons exerting pressure of 1000 psi with pressure gauges
Camera	Digital camera G10 (from Canon); resolution 640 x 480 pixels, 4 GB memory card

Table 2.5. Acquisition rates for the seismic, pressure and camera data collected during the experiments

Data	Rate of acquisition
Seismic	10^6 S/s
Pump Pressure	100 S/s
Well Pressure	10^6 S/s
Camera	30 frames/s

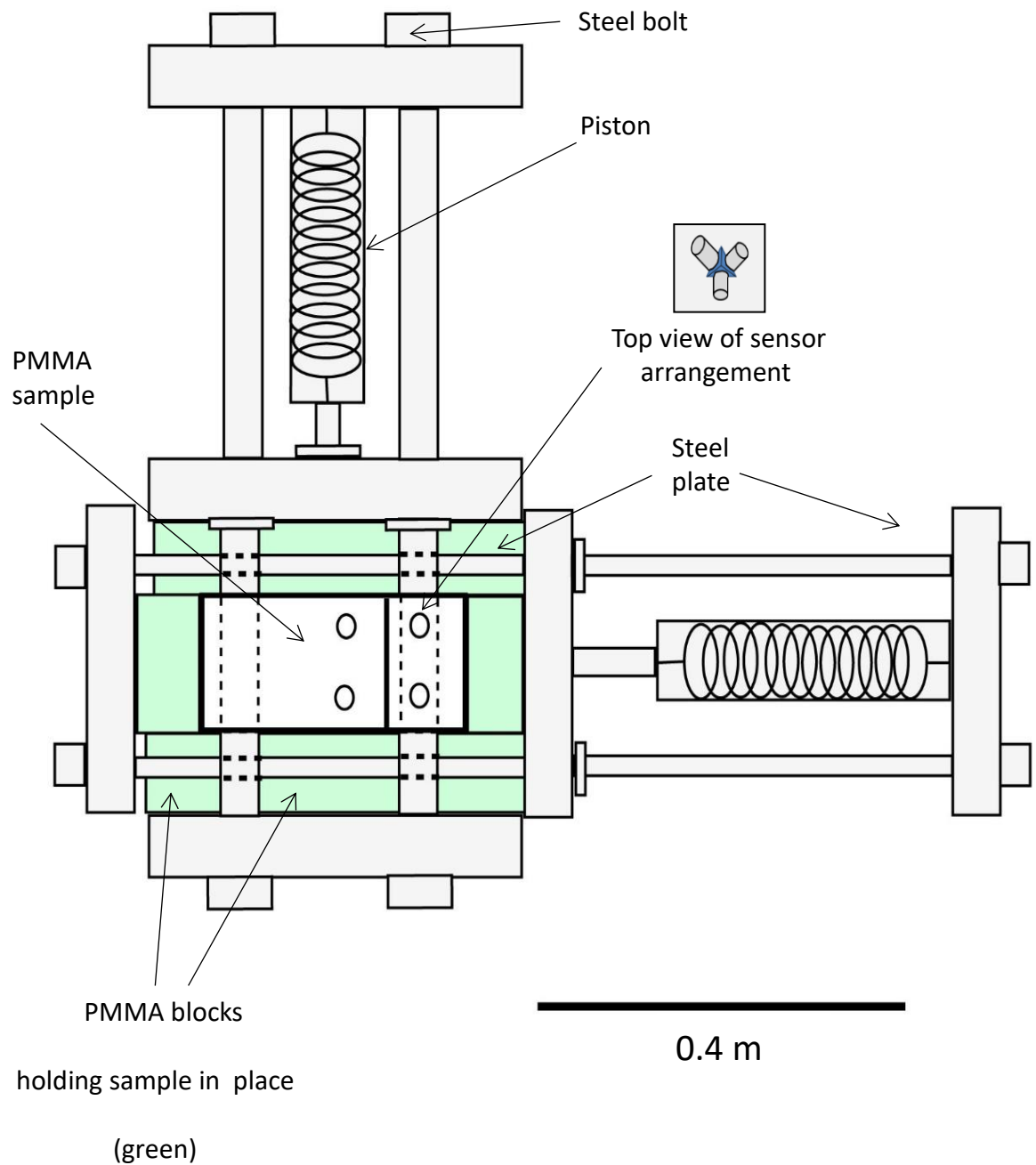


Figure 2.4. Diagram of the biaxial press containing the sample. The green shaded blocks represent the PMMA blocks that help to hold the sample in place during the experiment. The sample (in white) is in the middle of the press with 4 sensors seen on the top of the sample. We adjust the four steel plates in the middle of the press to secure the sample in the press. We set the pistons and measure the pressure using the gauges to ensure that the press exerts 1000 psi of pressure in the two horizontal directions.

Throughout the course of both experiments, we collect three main types of data- seismic, pressure and camera. In addition to the seismic, pressure and camera data, we take detailed notes of significant pressure variations, changes in the crack or sample, and record the local time throughout the experiment.

Each PMMA sample has 8 station mounts, each with 3 sensors connected in a Galperin arrangement to measure the displacement caused by the seismic events (Galperin, 1955; Graizer, 2009). The three component Galperin system of the sensors allows for easy distinction of noise from the signal and each sensor responds to gravity identically (Graizer, 2009; Townsend, 2014). We record the seismic data on three data acquisition cards and each card can collect 8 channels of data. The data collected on each card is slightly delayed because each card has its own independent internal clock and the cards are not synchronized.

The pressure data are collected at the pump and at the well to observe any differences that may occur in the timing of the pressure responses and for Nolte-Smith (Nolte and Smith, 1981). To observe the growth of the crack, we collect color video images from the digital camera so that we can observe what is happening to the fracture as it initiates and propagates through the sample.

Since we are collecting data on 3 different acquisition systems, we need to synchronize the data collection between each system. Our synchronization signals consist of a red, light-emitting diode (LED) in the field of view of the camera and voltage pulses. We inject synchronization signals into the pressure data, seismic data and video recording every 5 to 15 minutes apart during the experiments. The variability in the occurrence and length of each synchronization signal helps us to develop a common time frame for data collected in the experiments (Appendix).

2.4 Seismic data processing

The seismic data processing consists of identifying seismic events, preparing the data for analysis, obtaining the signal-to-noise ratios (SNR), dominant frequency, and magnitude, locations. RStudio is the integrated development environment for the open source programming software R, that we use to write and run programs to analyze the seismic data (R Core Team, 2013; RStudio Team, 2013). RHFM (Lorenzo, 2015) and RSEIS (Lees, 2014) are the main R-packages that we use in the processing of the microseismic data (Appendix for use in programs). Seismic Un*x is a seismic data processing open source programming software that we use in the seismic data analysis (Cohen and Stockwell, 2013). The program we use from Seismic Un*x is supolar (Maercklin, 2001), which we use for the principal component analysis (PCA) of the seismic data (Appendix for use in programs).

2.4.1 Microseismic event selection and Event characteristics

We pick events initially that have amplitude values greater than 80 counts because this is the noise level for the experiments. We further sort the events by visually identifying displacements and these events all have maximum amplitude values greater than 100 counts. We use cross-correlation to correct these delays in the seismic data when we collate the data. After cross-correlation of the events, we rotate the data into the principal component system-North, East and Up. We record the maximum amplitude and the average signal-to-noise ratio of each event.

We generate the amplitude spectrum of each event in order to identify the dominant frequency contained in the event by examining the seismic event in the frequency domain. The Fourier transform converts a continuous signal in the time domain, to the frequency domain,

$$u(f) = \int_{-\infty}^{\infty} u(t) e^{i2\pi ft} dt, \quad 2.5$$

where $u(t)$ is the seismic signal in terms of time and $u(f)$, is the transformed signal in terms of frequency (Liner, 2004).

Since our seismic signal consists of samples collected at a fixed sample rate (10^6), our seismic signals are not continuous so we approximate the integral of the Fourier transform by a summation, called the discrete transform,

$$g(f) = \sum_{n=1}^{nt} g(t_n) e^{i2\pi f t_n}, \quad 2.6$$

where nt is the number of samples in the signal, t_n , that we transform to the frequency domain (Liner, 2004). We use the fast Fourier transform because it computes the discrete transform in a shorter time frame by computing the operation $n \log_2 n$ times instead of n^2 without changing the results of the transform (Liner, 2004). The FFT gives us values split into real and imaginary numbers. The amplitude is the modulus of the complex number at each frequency (Liner, 2004).

The amplitude spectrum shows which the components of the frequency that the signal contains and gives a measure of how strong each on the frequency component are in the signal. For the amplitude spectrum we only use frequencies up to the value of half of the sample rate. This value gives the Nyquist frequency which is the highest frequency that can reliably represent the signal transformed by the FFT (Liner, 2004). Examining frequencies only up to the Nyquist values can also decrease any distortion of the frequency content by aliasing where our sample rate may be too large to adequately represent the seismic signal (Liner, 2004). We identify the dominant frequency for each component of every event, and find the average of these frequency to obtain the average dominant frequency of each event.

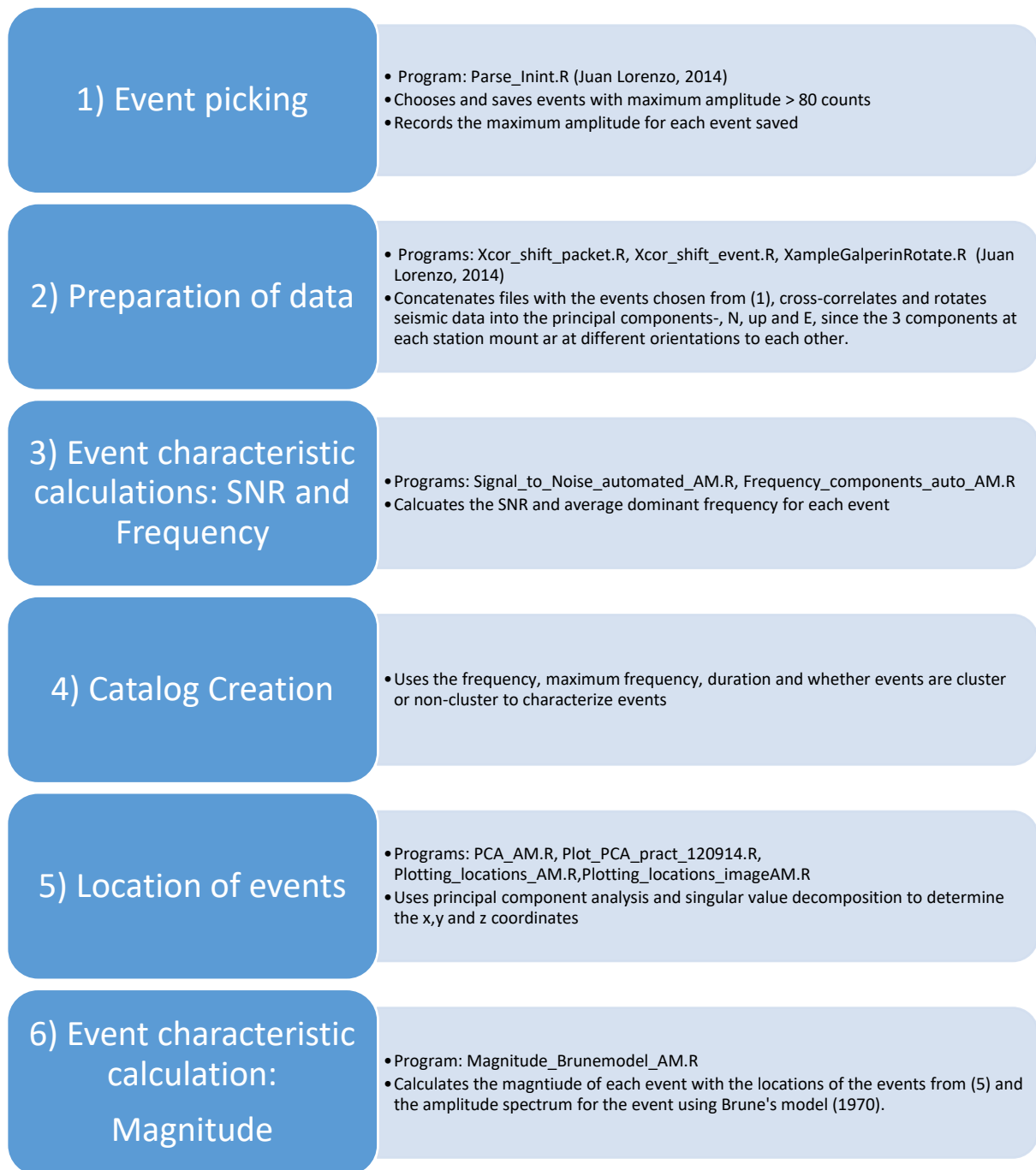


Figure 2.5. Flowchart of the steps in seismic processing: (1) We pick the seismic events above a minimum value of displacement of 80 counts and then, we keep events whose displacement we can pick above the noise level. (2) We cross correlate the data to align events since each acquisition card has an independent clock. (3) We calculate the signal-to-noise ratio and dominant frequency of each event. (4) Using the event characteristics from (3), duration, whether the event occurs singly or in a group, we classify the events. (5) We find the location of the events by using principal component analysis and a back projection to the source of the seismic events, compare the locations with the camera images and identify where each type of event occurs. (6) We calculate the magnitude of each event. The programs in step 3,5 & 6 are in the Appendix.

From the amplitude spectrum, we obtain the seismic moment, M_0 , for the microseismic events by using Brune's model for a circular crack where,

$$M_0 = \frac{4\pi\rho v^3 r \Omega_0}{R_p} \quad 2.7$$

where ρ is the density of the material, v is the P wave velocity, r is the source-receiver distance, Ω_0 is the frequency level of the amplitude spectrum, and R_p is the P wave radiation pattern correction term (Brune, 1970; Stork et al., 2014). We use the locations that we calculate in step 5 (Figure 2.5) to determine the source-receiver distance, r , and we estimate the frequency level, Ω_0 , from the low frequency plateau of the amplitude spectrum of the microseismic event (Baig and Urbancic, 2010).

For our experiments, since PMMA is considered an isotropic medium, we assume our predominant seismic waves as observed in our seismograms to be P waves. We expect most of the energy from the direct wave to arrive at the sensor since the samples are small and made of a homogenous material. We use $R_p = 0.52$, which is the average value for the correction term for P waves and can be applied to PMMA since the radiation pattern correction term is dependent on the take-off angle of the seismic wave, the azimuth from the seismic source and the focal mechanism and not on the material (Boore and Boatwright, 1984).

From M_0 , the magnitude of the microseismic event can be calculated from the relationship,

$$M_w = \frac{2}{3} \log_{10} M_0 - 6 \quad 2.8$$

where M_w is the moment magnitude of the event (Baig and Urbancic, 2010).

2.4.2 Location of Events

2.4.2.1 Principal Component Analysis (PCA)

Principal component analysis of three component data allows us to obtain the direction of the source that generates a microseismic event. The particle motion caused by a microseismic event, can be fit to an ellipsoid composed of the signals from each component (X, Y, and Z) of the stations for a given time window (Benhama et al., 1988; Maercklin, 2001). PCA uses a covariance matrix approach where for the three-component data- X, Y, Z- we compute the covariance matrix represented by

$$M = \begin{pmatrix} Var(X) & Cov(X,Y) & Cov(X,Z) \\ Cov(Y,X) & Var(Y) & Cov(Y,Z) \\ Cov(Z,X) & Cov(Z,Y) & Var(Z) \end{pmatrix}, \quad 2.9$$

$$Cov(X,Y) = \frac{1}{N} \sum_{k=-L}^L [X_k(t) - \mu_x][Y_k(t) - \mu_y] \quad 2.10$$

$$Var(X) = Cov(X,X) \quad 2.11$$

where N is the number of samples for half the chosen time window length, $L = \frac{N-1}{2}$, and μ is the mean value of each time sequence analyzed in the window (Maercklin, 2001).

The covariance matrix aims to maximize the variance of the components that is how different the components are from each other and minimize the covariance that is how closely related each component is to the other (Richardson, 2009). The maximization of the variance ensures that in fitting the ellipsoid, the most variation of the energy of the signal is accounted for by the principal (largest) component (Abdi and Williams, 2010). To achieve the maximum variance (where the covariance = 0, or ~ 0), we diagonalize the covariance matrix, M (Richardson, 2009). The resultant eigenvalues (or singular values) and eigenvectors

represent the maximum variance of each component of the signal which gives us the best fit of the polarization ellipsoid of the signal (Richardson, 2009).

The eigenvectors and eigenvalues of the covariance matrix, M , satisfy the following equation,

$$MV_i = \lambda_i V_i \quad 2.12$$

where V_i is the i^{th} eigenvector of M and λ_i is the i^{th} eigenvalue of M .

The eigenvectors, V_1 , V_2 and V_3 with their associated eigenvalues, λ_1 , λ_2 , and λ_3 , define the principal axes of the polarization ellipsoid. Eigenvector, V_1 , represents the principal axis of the polarization ellipsoid (Maercklin, 2001). For the analysis of our events, we use the quadratic resultant for the energy distribution which describes the average energy for the given time window of N samples (Maercklin, 2001).

From the PCA, we obtain the azimuth and dip of the direction of the principal axis of the polarization ellipsoid that in turn represents the direction of the source of the microseismic event. The azimuth, ϕ , is the angle between the principal axis of the polarization ellipsoid and the XZ plane while the dip, θ , is the angle between the principal axis and the horizontal plane (Maercklin, 2001),

$$\theta = \cos^{-1}(|z|), \text{ for } 0 \leq \theta \leq 90^\circ \quad 2.13$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right), \text{ for } 0 \leq \phi \leq 360^\circ \quad 2.14$$

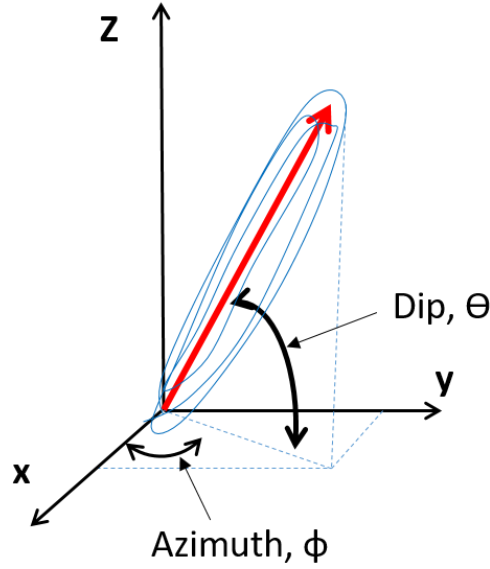


Figure 2.6. Illustration of the particle motion of the displacement that occurs during a hypothetical seismic event (thin blue line). The best fit polarization of the particle motion has the eigenvectors (V_1, V_2, V_3) as its axes (represented in red) with the longest axis representing V_1 . The azimuth, ϕ , and dip, Θ , are illustrated by the double-headed curved lines (Saenger et al., 2009).

2.4.2.2 Application of PCA to data

For every microseismic event for the two experiments, we carry out PCA to obtain values of the azimuth and dip of the projection of the line representing the direction of the source of the microseismic event

We plot all the azimuth and dip angles that we obtain from the PCA of a particular event (Figure 2.7a & b). We use four values (angles 1 to 4) each for the azimuth and dip of an event at each station. We pick the angle corresponding to the point of maximum energy in the time window as angle 1 (Figure 2.7 & 2.8). We then calculate the standard deviation for the time window chosen and angle 2 is equal to angle 1 plus the standard deviation while angle 3 is equal to angle 1 minus the standard deviation. The last angle we use, angle 4, is the average value in the chosen time window for the azimuth and dip respectively. The inclusion

of these four values for both azimuth and dip accounts for the maximum variation in the azimuth and dip that is caused by noise.

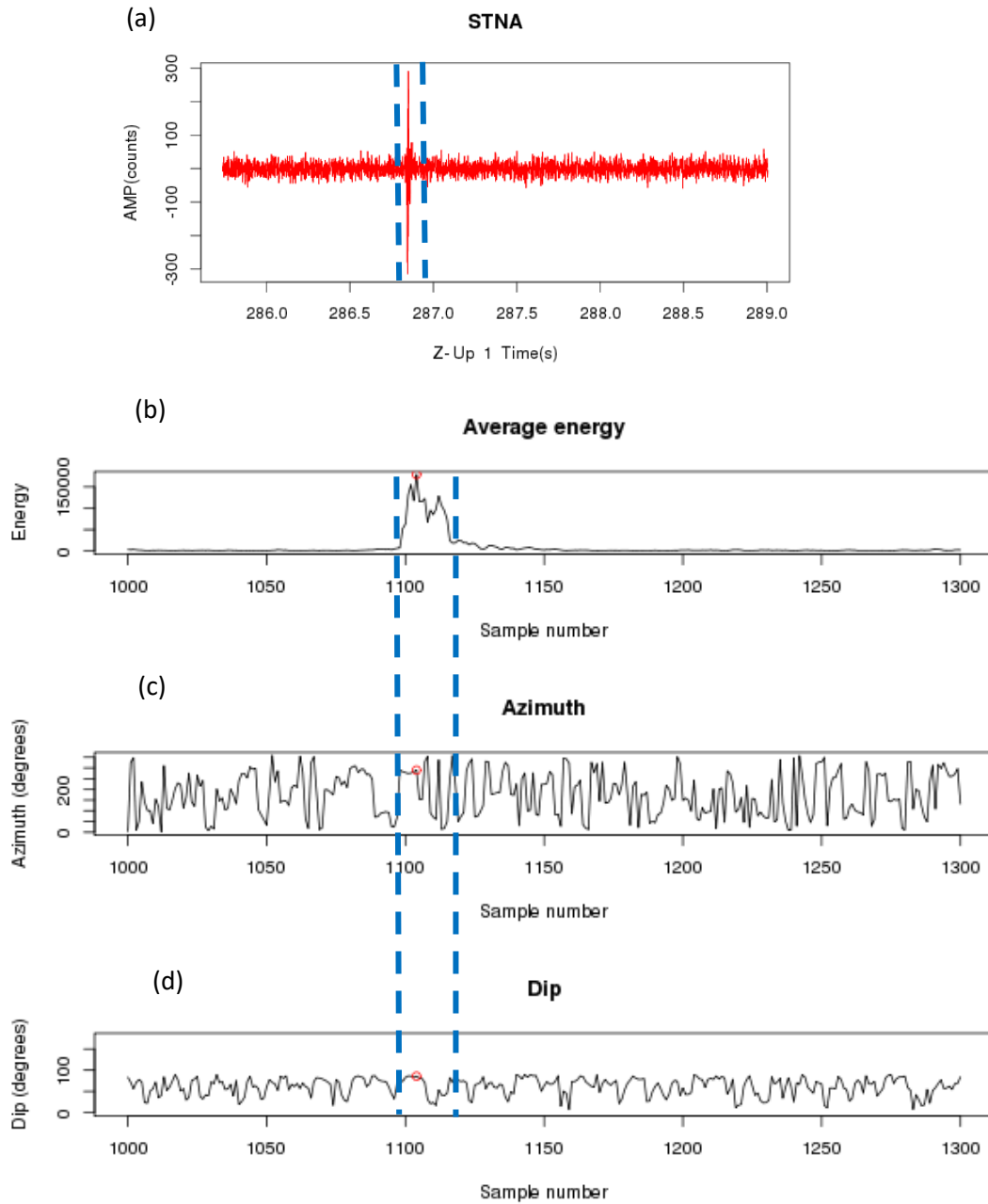


Figure 2.7. Principal component analysis of Event 1 from experiment 2 (Figure 3.1). The red circles represent the maximum value of the energy of the event (b), the corresponding azimuth and dip associated with the maximum energy (c and d respectively). We observe the variability of the azimuth and dip of the direction of motion of the seismic waves which is due to the influence of the noise.

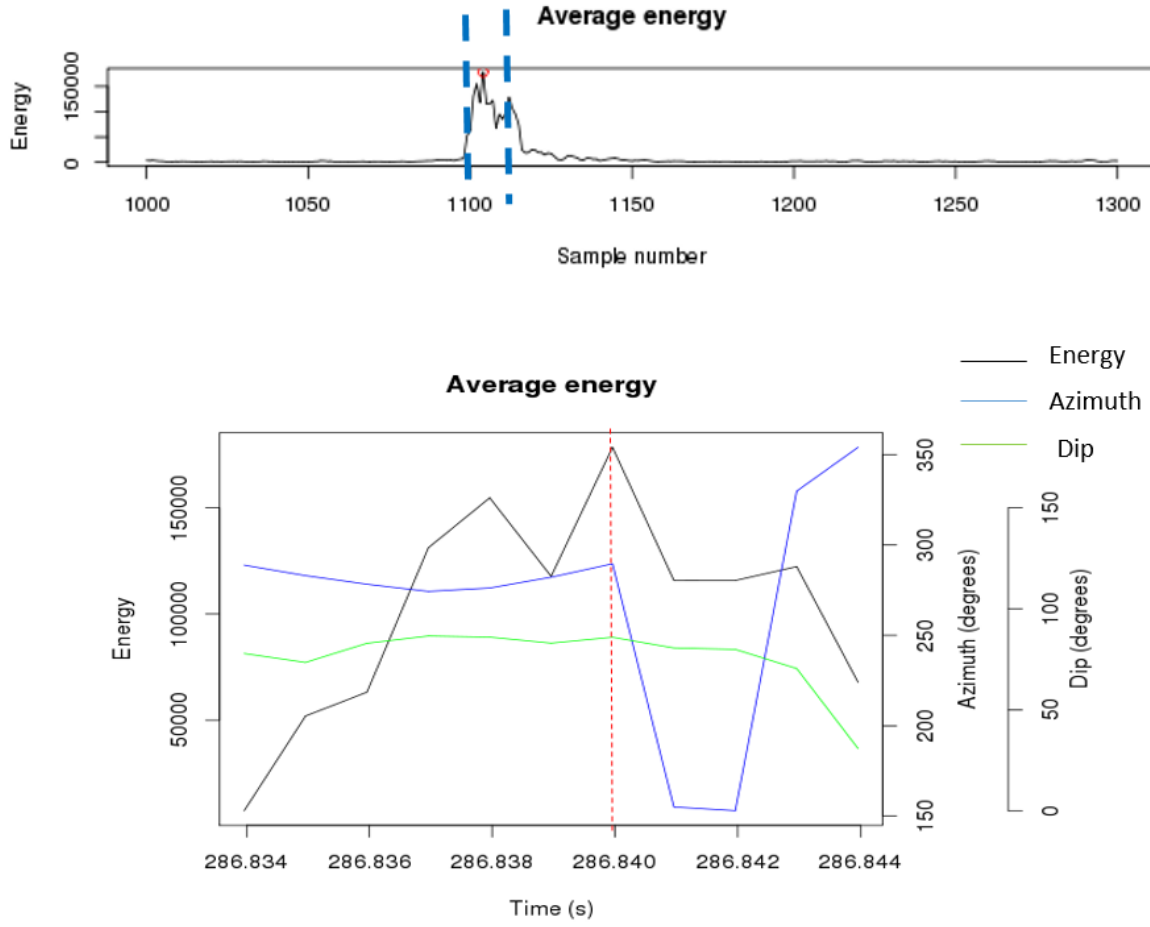


Figure 2.8. Magnification window of the maximum energy (black) for the event showing azimuth (blue) and dip (green) for the given window as well. Angle 1 for the azimuth is 290 and angle 1 for the dip is 86 (where the dotted red line intersects the blue and green lines respectively). Within this window, we find the standard deviation for the azimuth and dip and the average azimuth and dip angles.

2.4.2.3 Back Propagation for locating the microseismic events

We use back-propagation to determine the location of microseismic event (Han et al., 2010). The angles obtained from the PCA can be represented as a line (a direction vector) which traces the propagation of the P-wave from each seismic station back to the microseismic source (Han et al., 2010). We assume that ray paths are straight between the seismic source and there is no refraction. The point of intersection of the ray paths or the nearest point to all the ray paths represents the source location.

2.4.2.3.1 Nearest point to lines using Singular Value Decomposition

For our laboratory data, we use a simple matrix approach to determine the intersection of these lines or the closest point to the lines if all the lines do not intersect (Han et al, 2010). We use the Singular Value Decomposition (SVD) method to find the hypocenter locations from microseismic data for lines that intersect at a single point, are skewed and are parallel.

The x, y, and z coordinates of points on each line are obtained using the relationship of the azimuth and dip to the direction cosines:

$$x_2 = D * \cos \phi * \sin \theta + x_1 \quad 2.15$$

$$y_2 = D * \cos \phi * \cos \theta + y_1 \quad 2.16$$

$$z_2 = -D * \sin \theta + z_1, \quad 2.17$$

where x_1, y_1, z_1 , are the cartesian coordinates of the station, S_i , (receiver), x_2, y_2, z_2 , is a point along the line of the direction from source to receiver, D is the length of the line between the two points defined as $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$, ϕ is the azimuth and θ is the dip that we obtain from principal component analysis.

We have 8 stations for the experiment but we notice 3 stations, station B, C & D, have components with very low average signal-to-noise ratios (≤ 1). These stations for the two experiments have sensors with noisy data. We choose the 5 stations with the best signal-to-noise ratios and with all three components with working sensors for the calculations. We need to use stations with clear data on the three components since principal component analysis requires the use of the 3 components.

The coordinates $S_n(x_n, y_n, z_n)$ represent each station, where S_n is a particular station used and n goes from 1 to 5 and the direction vectors representing the stations are defined as $U_n(u_{xn}, u_{yn}, u_{zn})$, where U_n represents the direction vector from station and S_n to the coordinates of the location of the event (Han et al.,2010).

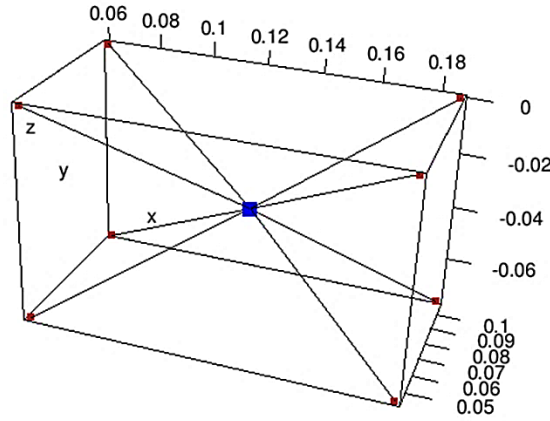


Figure 2.9. Diagram of an SVD test of 8 lines with known equations. The black lines within the box represent the input equations, the red squares represent the station locations in this example, and the blue square represents of solution of the SVD technique which correctly gives the intersection point of the 8 lines used.

Using the equation for a line in 3D space, each line of the propagation path of the P-wave from the microseismic source to the receiver is

$$\frac{x - x_n}{u_{xn}} = \frac{y - y_n}{u_{yn}} = \frac{z - z_n}{u_{zn}} = d_n \quad 2.18$$

where d_n represents Euclidean length along each of the lines respectively (Han et al., 2010).

We can then expand the system of equations shown above in terms of x_n, y_n, z_n for each line

for n from 1 to 5:

$$\begin{cases} x + 0.y + 0.z - u_{x1}.d_1 - 0.d_2 \dots - 0.d_n = x_1 \\ 0.x + y + 0.z - u_{y1}.d_1 - 0.d_2 \dots - 0.d_n = y_1 \\ 0.x + 0.y + z - u_{z1}.d_1 - 0.d_2 \dots - 0.d_n = z_1 \\ x + 0.y + 0.z - 0.d_1 - u_{x2}.d_2 \dots - 0.d_n = x_2 \\ 0.x + y + 0.z - 0.d_1 - u_{y2}.d_2 \dots - 0.d_n = y_2 \\ 0.x + 0.y + z - 0.d_1 - u_{z2}.d_2 \dots - 0.d_n = z_2 \\ \vdots \\ x + 0.y + 0.z - 0.d_1 - 0.d_2 \dots - u_{xn}.d_n = x_n \\ 0.x + y + 0.z - 0.d_1 - 0.d_2 \dots - u_{yn}.d_n = y_n \\ 0.x + 0.y + z - 0.d_1 - 0.d_2 \dots - u_{zn}.d_n = z_n \end{cases} \quad 2.19$$

These equations can then be represented in matrix form:

$$Gm = X \quad 2.20$$

Where, G is a 15 by 8 matrix as shown below:

$$G = \begin{bmatrix} 1 & 0 & 0 & -u_{x1} & 0 & & 0 \\ 0 & 1 & 0 & -u_{y1} & 0 & \dots & 0 \\ 0 & 0 & 1 & -u_{z1} & 0 & & 0 \\ 1 & 0 & 0 & 0 & -u_{x2} & & 0 \\ 0 & 1 & 0 & 0 & -u_{y2} & \dots & 0 \\ 0 & 0 & 1 & 0 & -u_{z2} & & 0 \\ \vdots & & & & & & \\ 1 & 0 & 0 & -u_{x1}.0 & -u_{x2}.0 & \ddots & -u_{xn} \\ 0 & 1 & 0 & -u_{y1}.0 & -u_{y2}.0 & \ddots & -u_{yn} \\ 0 & 0 & 1 & -u_{y1}.0 & -u_{z2}.0 & \dots & -u_{zn} \end{bmatrix} \quad 2.21$$

and m and X are column vectors,

$$m = \begin{bmatrix} x \\ y \\ z \\ d_1 \\ d_2 \\ \vdots \\ d_5 \end{bmatrix}, \text{ and } X = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_5 \\ y_5 \\ z_5 \end{bmatrix}.$$

The singular value decomposition allows for the simple calculation of the minimization of

$|Gm - X|$, which gives the geometric solution of the nearest point to G. The vector m gives

the x, y and z values of the nearest point ($m[1:3]$) and the distances from each line to this point ($m[4:4+n]$).

The singular value decomposition of G (equation 2.18) gives:

$$G = U_{p \times p} * S_{p \times q} * V_{q \times q}^T \quad 2.20$$

where $p = m \times 3$ and $q = m + 3$, U is a $p \times p$ matrix, the columns of U are known as the left singular vectors of A (the orthonormal eigenvectors of $G G^T$), V is a $q \times q$ matrix with the columns of V called the right singular vectors of A (the orthonormal eigenvectors of $G^T G$), and S is a $p \times q$ diagonalized matrix containing the singular values of G which are the positive square roots of the eigenvalues of $G^T G$ (Klema and Laub, 1980; Han et al., 2010).

Replacing G with the SVD of G (equation 2.22) in equation 2.18 and rearranging equation 2.18 gives:

$$m = V_r * S_r^{-1} * U_k^T * X \quad 2.21$$

If $r = q$, as is the case for lines intersecting at a single point or skewed lines, then the nearest point to all the lines in 3D space is a unique solution,

$$Nearest\ point_{ALL} = m [1:3]. \quad 2.22$$

2.4.2.4 Location refinement and errors

For every event, there are 4 sets of azimuth and 4 sets of dip angles obtained from the PCA analysis of the seismic event. We also use 3 values for the locations of the stations- the measured value, and the upper and lower limits of the value including the error in measurement. We use 4^n combinations of the values for the azimuth and dip, where n is the

number of stations used in the back-projection, for the three values of the stations. We have 4^5 (1024) combinations when we use 5 stations.

We find the density of the points obtained from the combinations of values for x, y and z coordinates where,

$$f_n(x) = \frac{1}{nh_n} \sum_{i=1}^n k\left(\frac{x - X_i}{h_n}\right), \quad 2.23$$

$$f_n(y) = \frac{1}{nh_n} \sum_{i=1}^n k\left(\frac{y - Y_i}{h_n}\right), \quad 2.224$$

$$f_n(z) = \frac{1}{nh_n} \sum_{i=1}^n k\left(\frac{z - Z_i}{h_n}\right), \quad 2.235$$

where k is the kernel function, f_n is the function that is fit to the sample points (X, Y, Z) , h_n is the bandwidth (smoothing parameter for the function) (Chaubey et al., 2012).

The kernel function gives a measure of how influential each point in the groups of coordinates (X_i, Y_i, Z_i) is on the function, f_n , that is fit and smooth to the points. The points that are closest to each other have the greatest influence on f_n . The maximum point of the graph of f_n , gives us the coordinate around which the most points are clustered (Chaubey et al., 2012; Scott, 2015).

We keep the locations with x, y, and z values that fall within the top 10 % of the kernel density functions for each of the coordinates (Figure 2.10). To obtain the location of the event, we find the average of each of the coordinates. We use the average location as the center of an error ellipsoid that is fit to the remaining locations. The semi-axes of the ellipsoid represent the margin of error for each location.

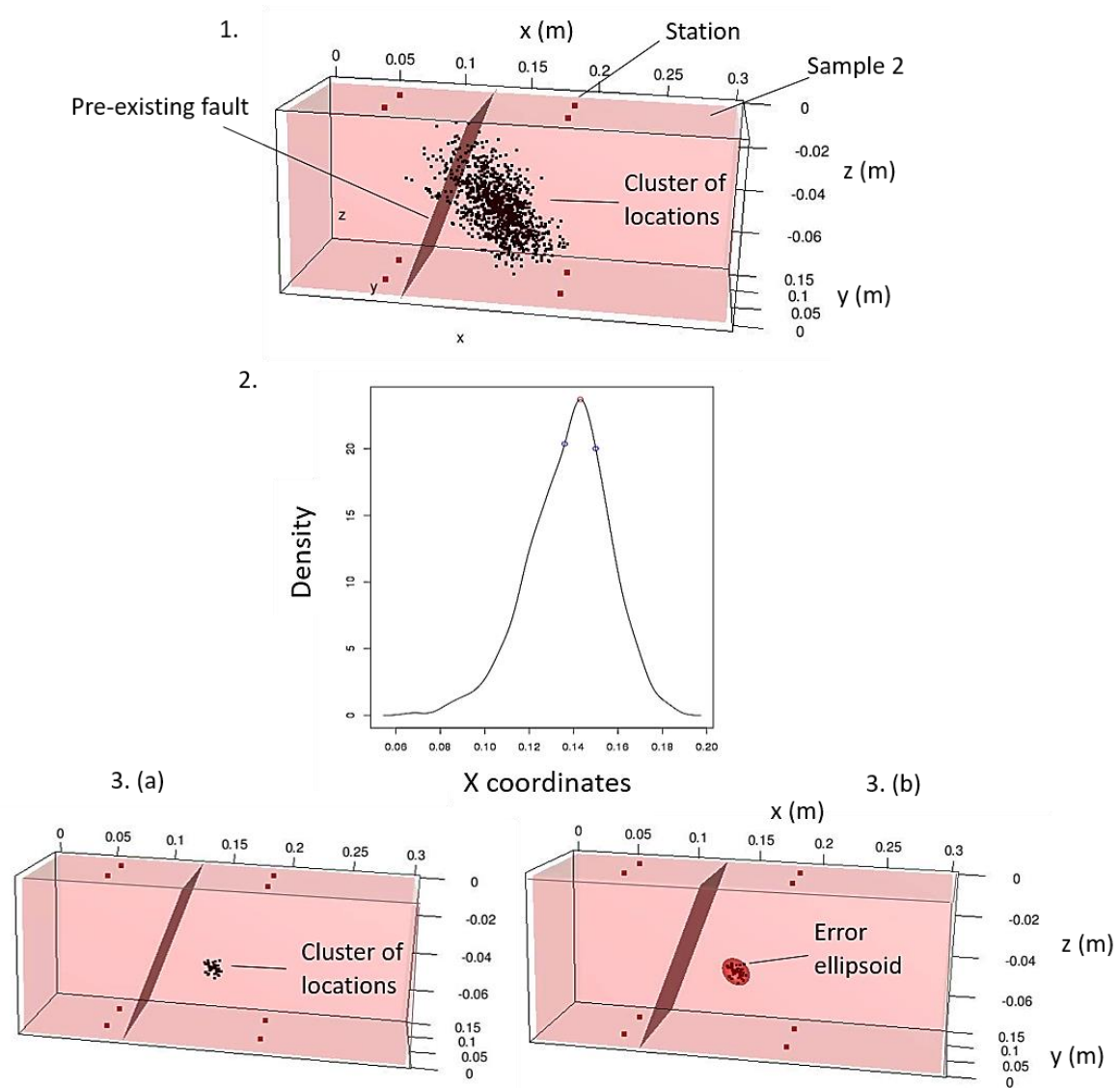


Figure 2.10. Illustrations of the steps in refinement of all of the possible locations for event 1 in experiment 2 using the SVD technique, kernel density function and error ellipsoid fitting. (1) shows the results of the SVD for all the combinations of the locations (1024) for each event on a schematic diagram of sample 2. (2) is a plot of the kernel density function for the x coordinates of all the possible locations. The peak in the graph corresponds to the coordinate with the greatest number of locations closest to it. (3a) shows the possible locations left (< 100) after using the kernel density function on x, y and z coordinates. (3b) shows an error ellipsoid fit to the remaining locations. The center gives the actual location we use and the axes of the ellipsoid to give us the errors in the location.

2.5 Pressure Data Analysis

We use the relationships between the fracture dimensions and the pressure response to help explain the variations we observe in the pressure data. We can consider the fracture's behavior in terms of the critical parameters affecting fracture growth and other parameters associated with the fracturing fluid namely pump rate, Q and fluid viscosity, μ . The critical parameters affecting fracturing are the height of the fracture, H ; Young's modulus, E ; the fluid loss coefficient, C ; fracture toughness, K_{IC} (Jones and Britt, 2009). For the purposes of the experiment, because PMMA is a non-porous material, we assume that all of the volume pumped into the samples is transferred to the volume of the fracture so the volume lost is negligible in the experiments.

The net pressure (P_{net}) is the difference in treating pressure and the closure pressure (σ_{cl}),

$$P_{net} = BHTP - \sigma_{cl} \quad 2.24$$

The net pressure is related to the parameters affecting fracture growth in the following equation,

$$P_{net} \approx \frac{E'^{\frac{3}{4}}}{L_1} [\mu Q L_2]^{\frac{1}{4}} + P_{tip}, \quad 2.25$$

where μ is the viscosity of the fracturing fluid, Q is the injection rate of the fracturing fluid, L_1 and L_2 are the dimensions of the fracture (Figure 2.12), and P_{tip} is the pressure at the tip of the fracture (Jones and Britt, 2009).

For the fracture growth that is not considered radial, that is where the ratio of L_1/L_2 , is not equal to 1, the effect of pressure near the tip of the fracture becomes more important and

thus P_{tip} term in equation 2.3 becomes more dominant. The pressure near the tip of the fracture is

$$P_{tip} = K_{IC} \sqrt{\frac{\pi}{24L_1}} \quad 2.26$$

and

$$K_{IC} = \Delta\sigma_1 \sqrt{\pi L_2} \quad 2.27$$

where K_{IC} is the fracture toughness and $\Delta\sigma_1$ is the differential stress (Jones and Britt, 2009).

We can substitute equation 2.24 and 2.25 into equation 2.23 to obtain,

$$P_{net} \approx \frac{E'^{\frac{3}{4}}}{L_1} [\mu Q L_2]^{\frac{1}{4}} + K_{IC} \sqrt{\frac{\pi}{24L_1}}, \quad 2.30$$

$$P_{net} \approx \frac{E'^{\frac{3}{4}}}{L_1} [\mu Q L_2]^{\frac{1}{4}} + \Delta\sigma_1 \sqrt{\pi L_2} \cdot \sqrt{\frac{\pi}{24L_1}}, \quad 2.28$$

$$P_{net} \approx \frac{E'^{\frac{3}{4}}}{L_1} [\mu Q L_2]^{\frac{1}{4}} + \Delta\sigma_1 \pi \sqrt{\frac{L_2}{24L_1}}, \quad 2.29$$

When our fracture grows radially, $L_1 \approx L_2$, and so

$$P_{net} \approx \frac{E'^{\frac{3}{4}}}{L_1} [\mu Q L_2]^{\frac{1}{4}} + \Delta\sigma_1 \pi \sqrt{\frac{1}{24}}. \quad 2.30$$

The abovementioned equations provide us with an idea of the expected or theoretical behavior of the pressure response during the growth of the hydraulic fracture. The net pressure and thus the pressure response from the pump changes as our fracture dimensions change since the fluid viscosity, the flow rate, the properties of the samples and the pressure applied are generally constant throughout the experiments. For instance, can see from equations 2.27 & 2.28, that an increase in the pump pressure, Q , should result in an increase in the P_{net} . In experiment 2, however, the flow rate changes twice during the experiment at specific times and we consider how these changes affect the pressure in experiment 2 (Figure 3.24).

2.6 Camera Data

We also record the time that these images represent in the crack growth in relation to our reference time frame. From the video images of the crack in both experiments, we calculate the area, circumference and lengths of the crack (Figures 2.11 & 2.12). ImageJ software calculates the dimensions of the crack produced in the two experiments. ImageJ is a software typically used in microbiology for scaling photographs and images of slides, thin sections associated with biological objects (Rasband, 1997). We use a known length on the image where the length is assigned a pixel value giving a scale in pixels per unit length (we use the lengths in centimeters for this software) (Rasband, 1997).

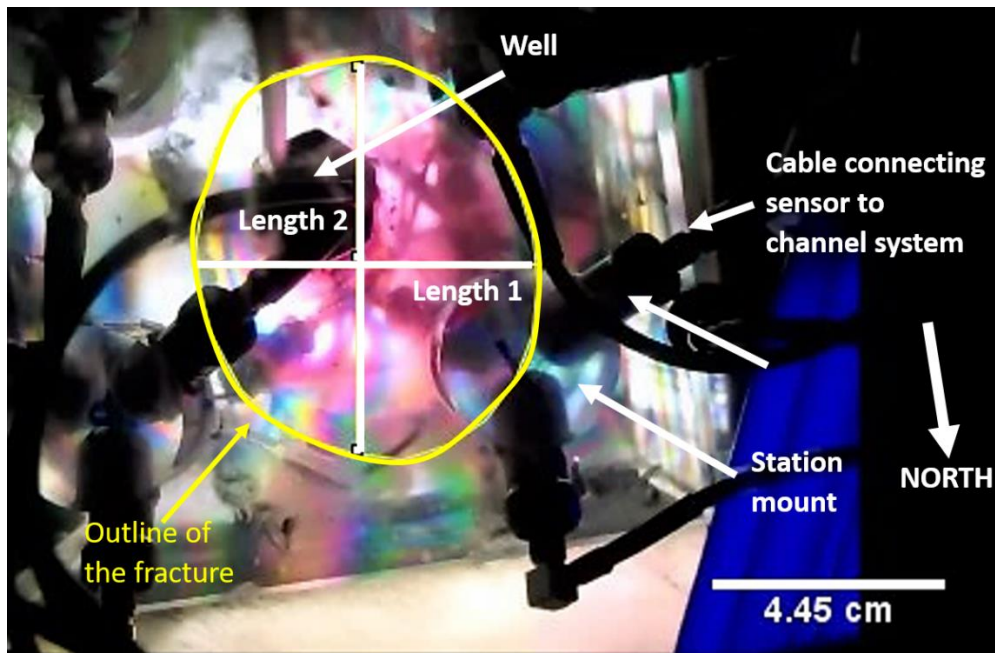


Figure 2.11. Image of the crack during experiment 1 at 146.47 minutes from the start of pumping at the beginning of the experiment. Length 1, length 2, the area and the perimeter of the crack are the measurements we take at every 2 to 3 minutes of recording from the beginning to the end of fracture growth.

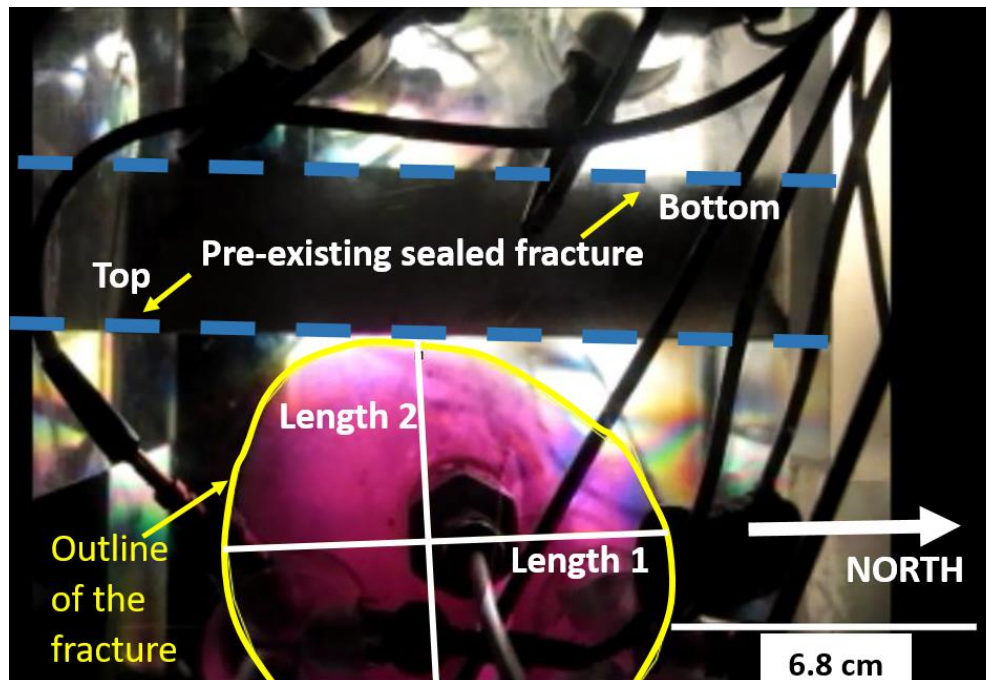


Figure 2.12. Image of the created fracture during experiment 2 at 268.29 minutes from the start of pumping at the beginning of the experiment. At approximately 138.63 minutes into the experiment, the field of view of the camera changes so that the eastern side of the created fracture is not completely visible. We estimate the size of the area cut off for the measurements of the fracture.

Chapter 3 Results and Interpretations

3.1 Microseismic event classifications

We classify the microseismic events from the two experiments by using the frequency, duration of the events, amplitude and whether the events individually or as a cluster (Table 3.1). The major categories of the microseismic events that occur in the two experiments (Table 3.1) are type I and II (Figures 3.1, 3.2 & 3.3), with high average dominant frequencies $>10,000$ Hz, and type III and IV (Figures 3.4, 3.5 & 3.6), with low average dominant frequencies of $<10,000$ Hz. The events can occur singly or within a cluster. A cluster is defined as a group of two or more microseismic events that occur within a 4 second window with no more than 0.5 seconds between events. Event types II (Figures 3.2 & 3.3) and IV (Figures 3.5 & 3.6) are high frequency and low frequency cluster events respectively where clusters with 2 identifiable events are grouped into type II-a (Figure 3.2) and IV-a (Figure 3.5), and clusters with more than 2 identifiable events are grouped into type II-b (Figure 3.3) and IV-b (Figure 3.6).

Overall, experiment 2 has a greater amount of microseismic activity than experiment 1. Experiment 2 has 110 identified events (Appendix F), which is ~ 7 times the number of identified events occurring in experiment 1, which has 15 identified events (Appendix E). The amplitudes of the microseismic events in experiment 2 are generally higher, with the highest amplitude reaching 32,552 counts as compared to 1,449 counts in experiment 1. Most of the microseismic events in both experiments are high frequency events type I and II. One of the 15 identified events in experiment 1 occurs as a low frequency type I event while experiment 2 has approximately 50 % of the 110 identified events classified as event types I and II. In both

experiments, the low frequency events have higher average amplitudes where event types III, IV-a and -b are 1290, 2375, 15790 counts greater than event type I, II-a and -b respectively.

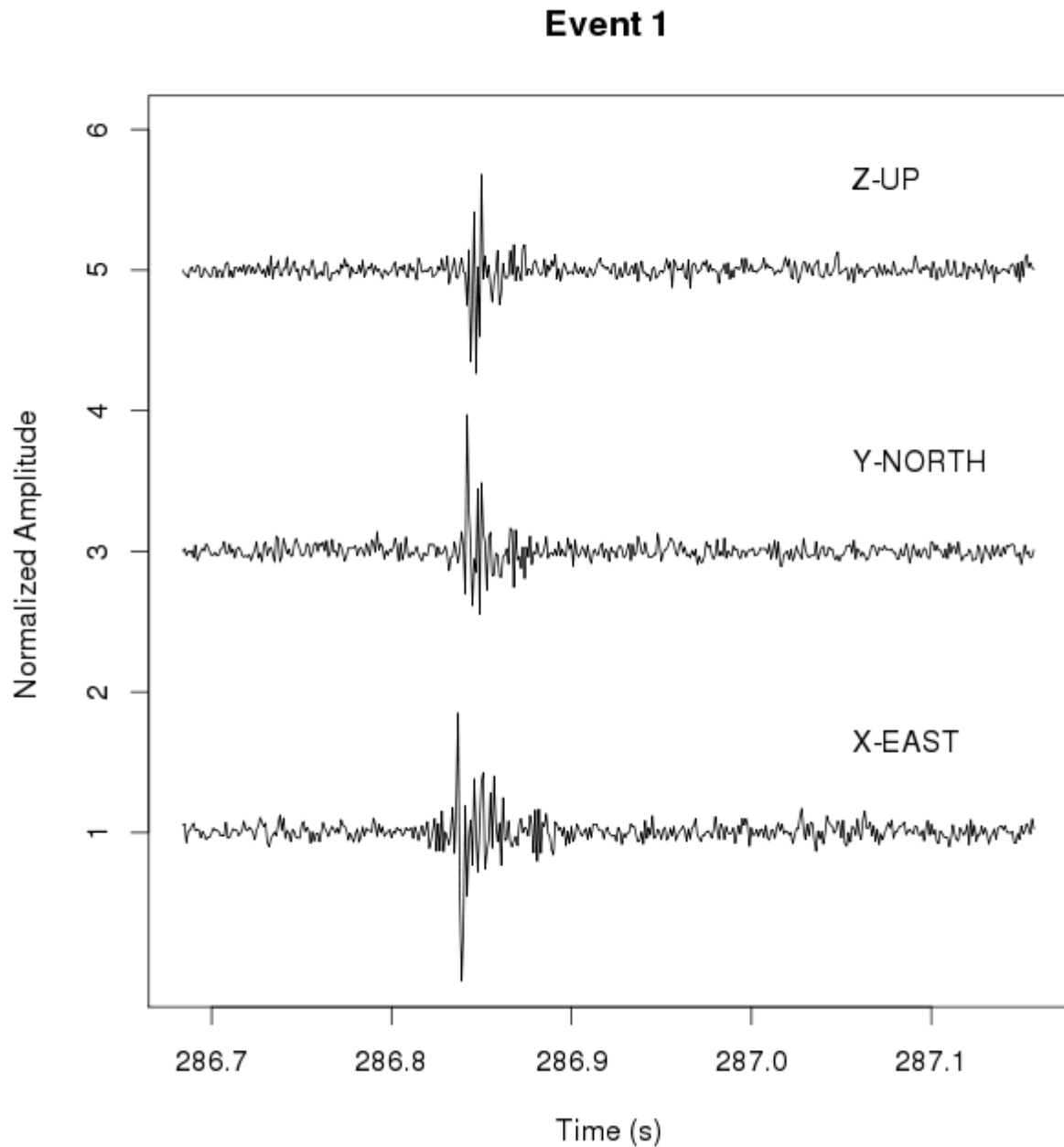


Figure 3.1. Microseismic event type I at station A, occurs as a single arrival with a frequency > 10,000 Hz lasting 0.5 seconds. The components, X, Y, and Z are in the principal coordinate system of north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.

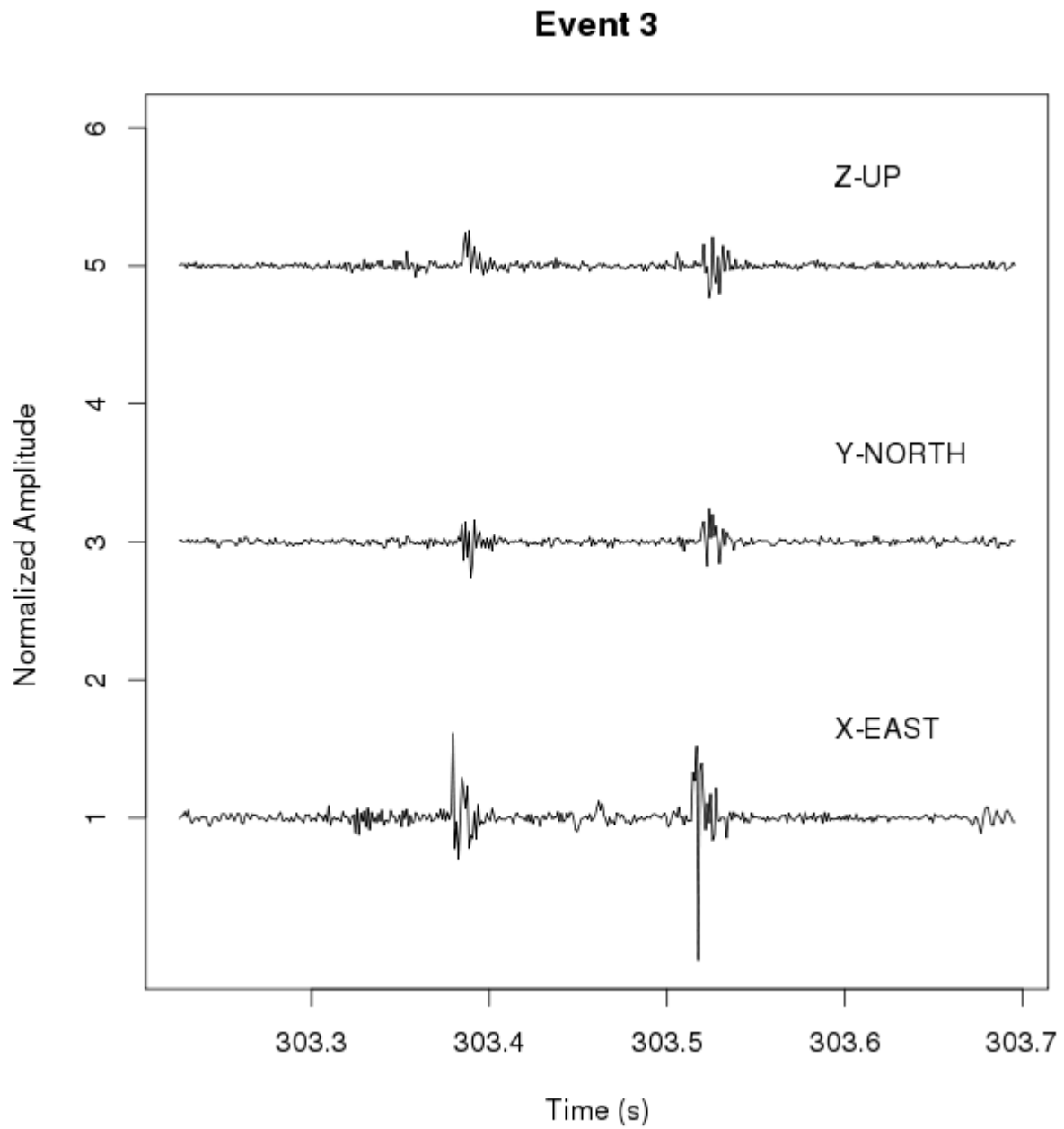


Figure 3.2. Microseismic event type II-a at station A occurs as a cluster of 2 events with an average dominant frequency $> 10,000$ Hz, lasting 0.5 seconds. The three components, X, Y and Z are in the principal coordinate system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.

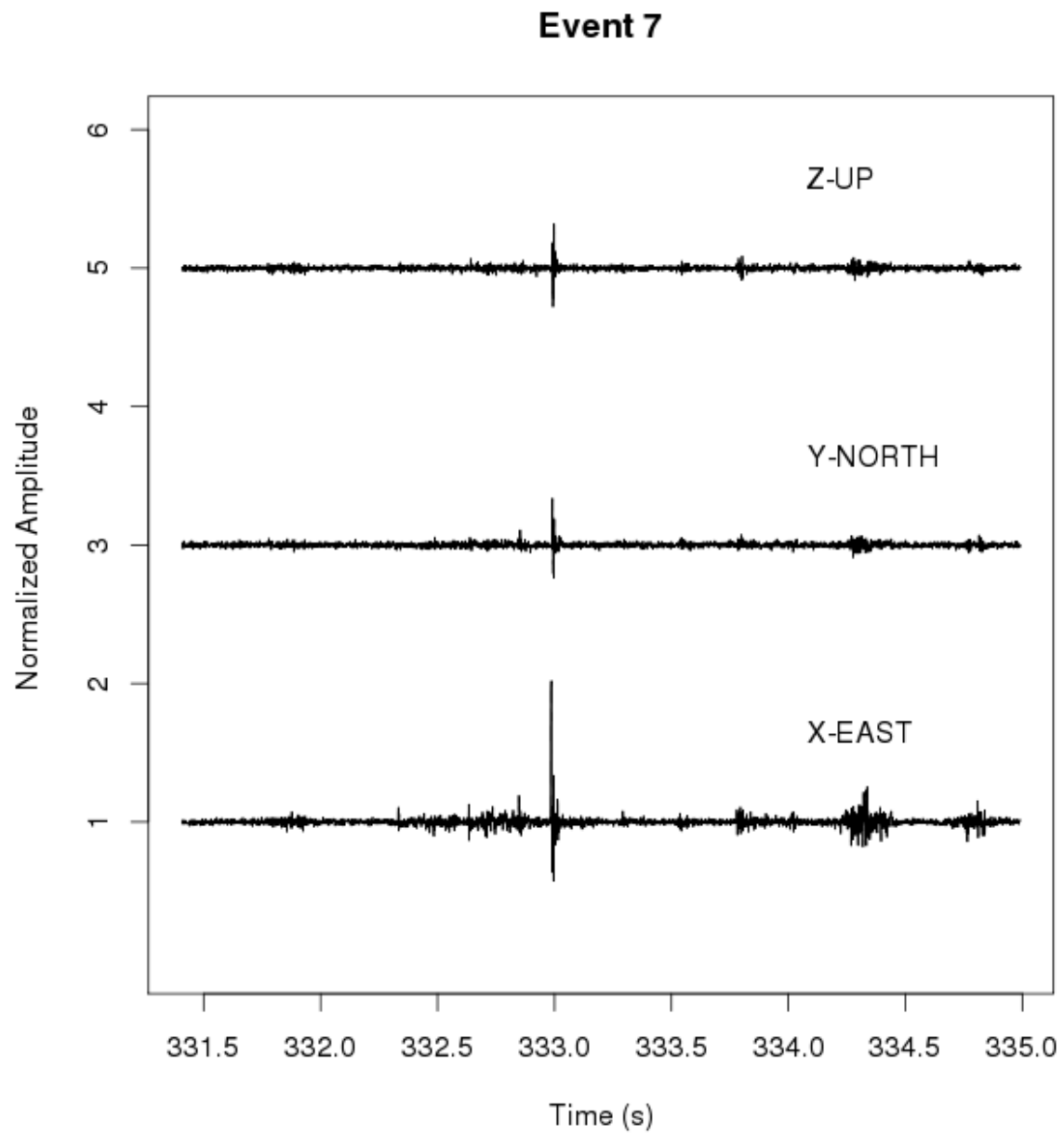


Figure 3.3. Microseismic event type II-b at station A with a frequency $> 10,000$ Hz occurs in a cluster with 5 identifiable arrivals. The 3 components X, Y, and Z, are in the principal system north, east and up.

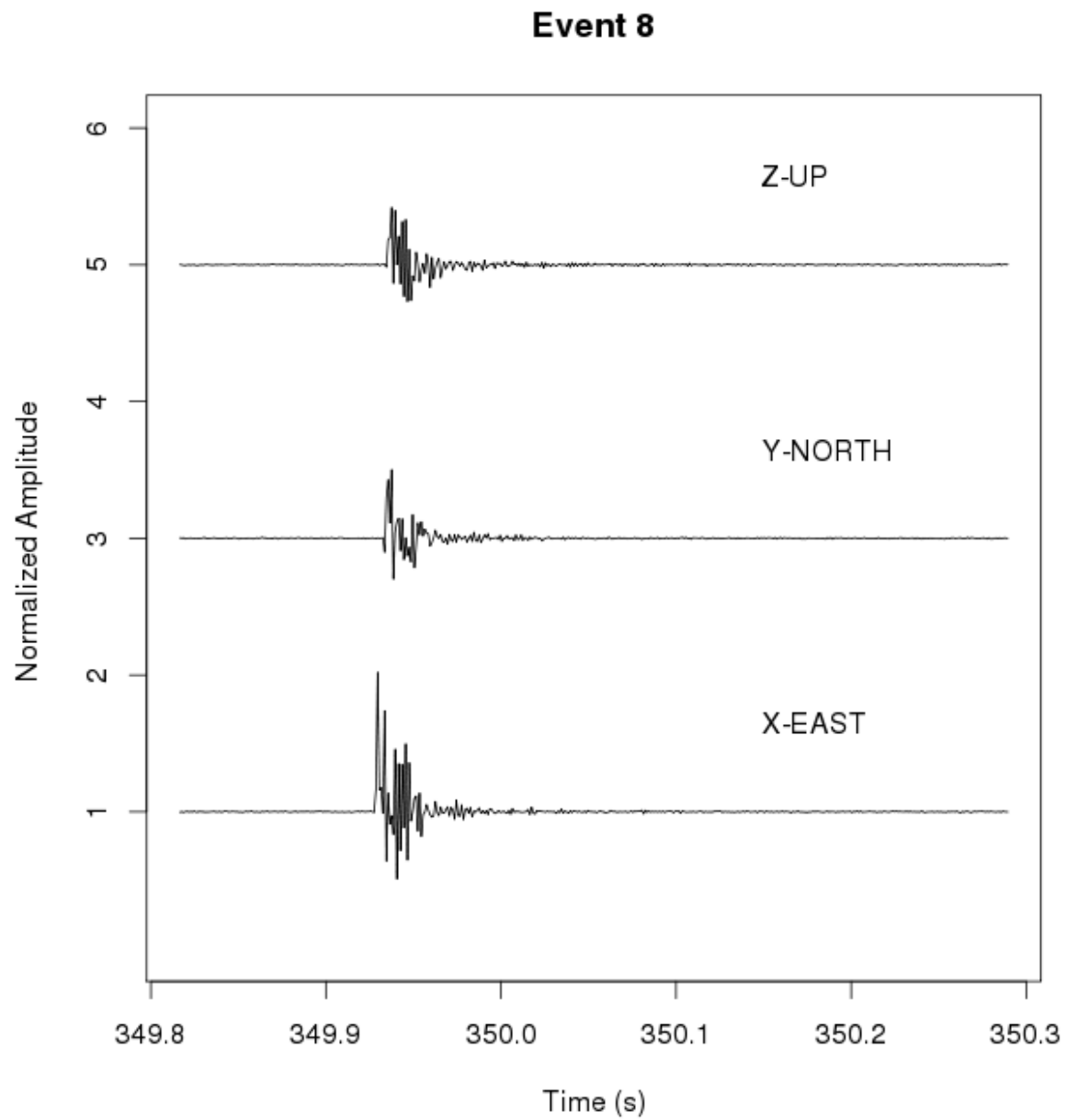


Figure 3.4. Microseismic event type III at station A with a frequency $< 10,000$ Hz occurs as a single event. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per s.

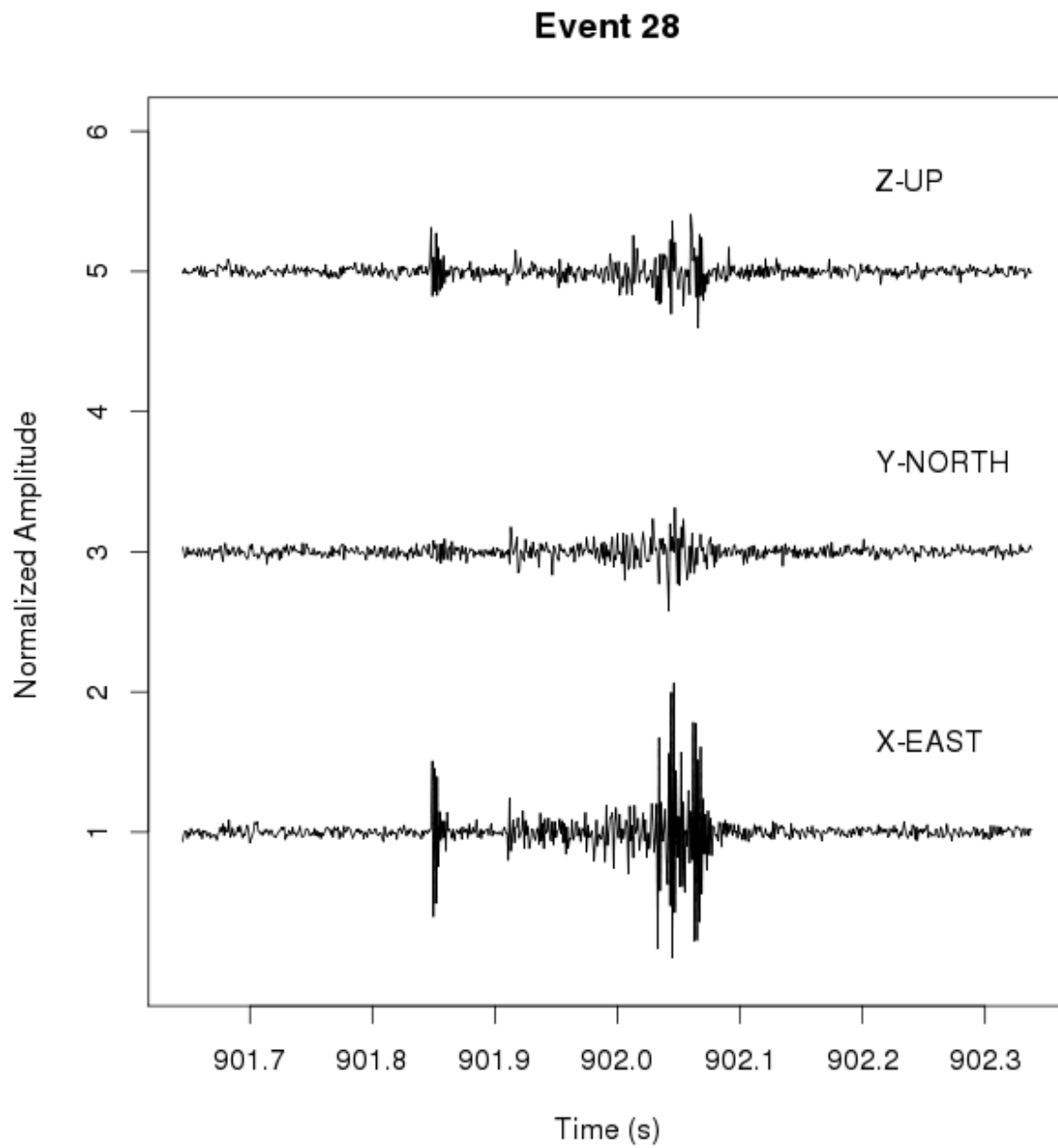


Figure 3.5. Microseismic event type IV-a at station A with a frequency $< 10,000$ Hz occurs as a cluster of 2 events. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per second.

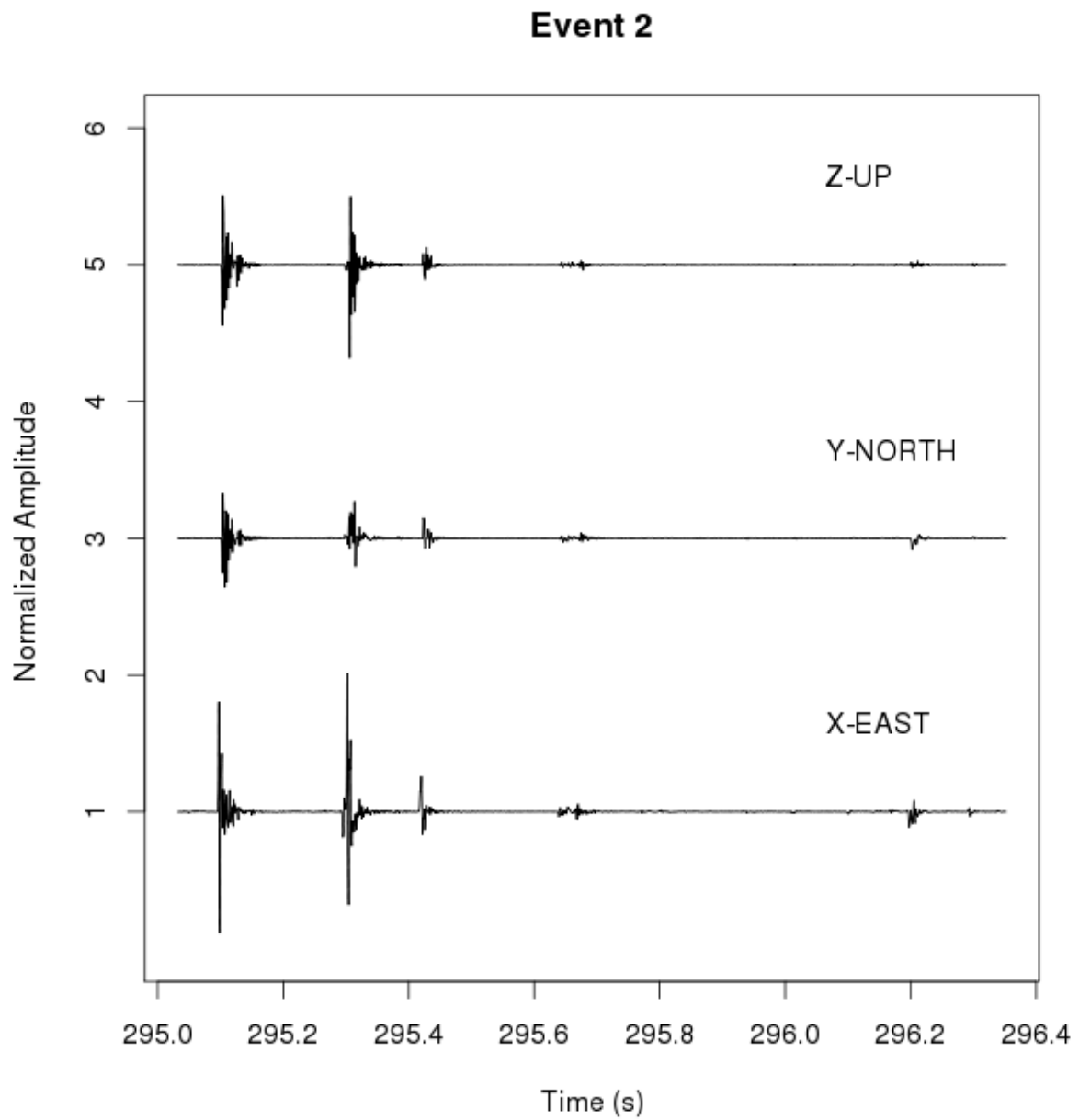


Figure 3.6. Microseismic event type IV-b at station A with a frequency $< 10,000$ Hz occurs as a cluster with more than 2 events. The 3 components X, Y, and Z, are in the principal system north, east and up. For a clearer view of the event, we decimate the data to 10^3 samples per second.

Table 3.1. Event classifications of the microseismic events identified in the two experiments. We classify the events based on the results of the two experiments using frequency, duration of the events, amplitude and whether the events are an individual event or a cluster of events.

Type	Frequency (Hz)	Avg Duration of individual event (s)	Amplitude (counts)	Cluster	Cluster		
					# of events	Avg Time between events (s)	Avg Duration full cluster
I	> 10,000	0.02-0.3 ~0.15	130-2900 ~421	NO	N/A	N/A	N/A
II-a	> 10,000	0.05-0.3 ~0.15	~425	YES	2	0.4	1
II -b	> 10,000	0.1-0.4 ~0.2	180-3100 ~610	YES	>2	0.5	4
III	< 10,000	0.01-0.35 ~0.3	140-16,200 ~1711	NO	N/A	N/A	N/A
IV-a	< 10,000	0.05-0.2 ~0.09	420-4500 ~2800	YES	2	0.2	1
IV-b	< 10,000	0.1-0.2 ~0.2	300 -32,500 ~16,400	YES	>2	0.4	4

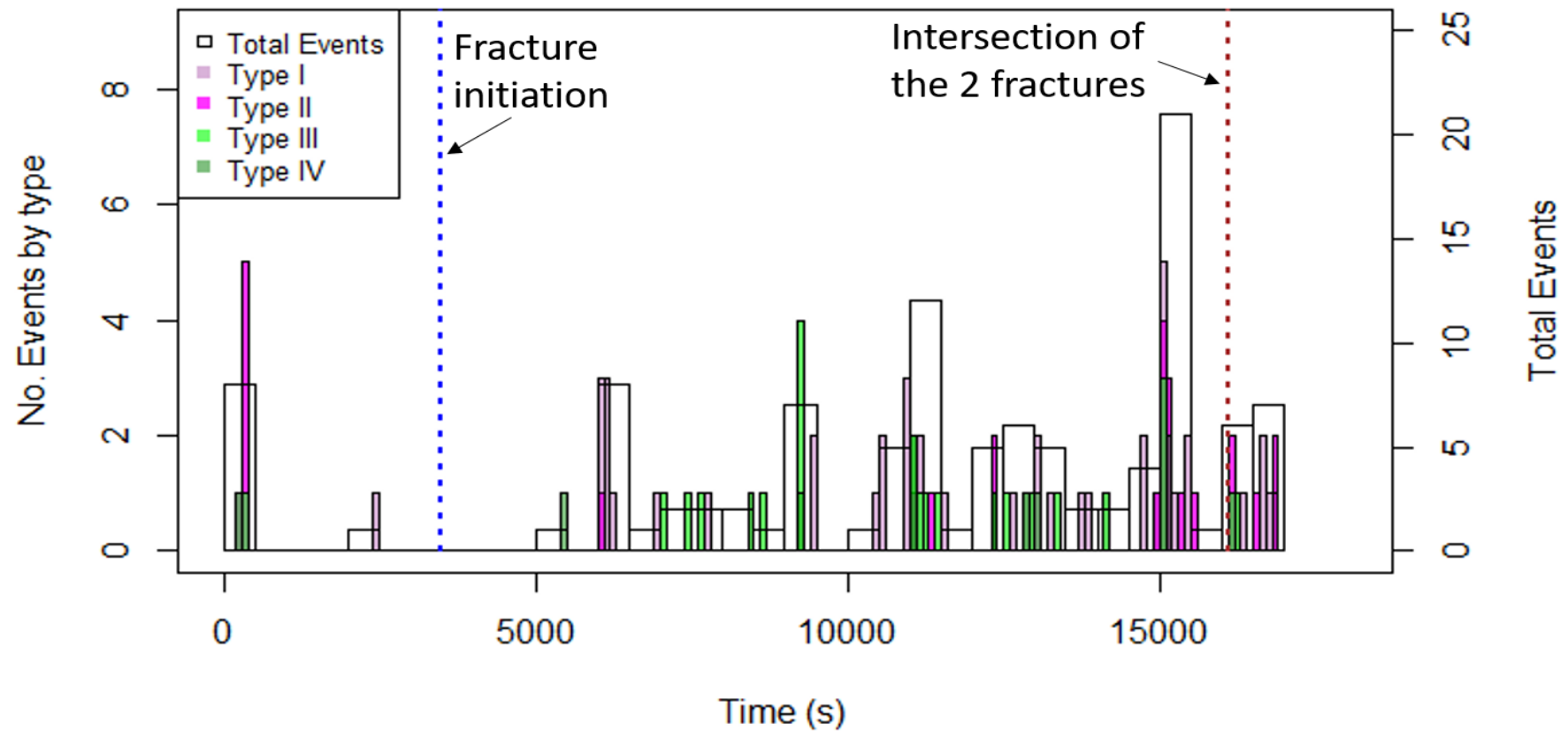


Figure 3.7. The distribution of the total number of microseismic events and the number of each event type throughout experiment 2. Most events occur between 5,000 and 20,000 seconds (during fracture growth), but we identify events that occur before fracture initiation. The events vary from low frequency to high frequency with no quantifiable pattern observed between the occurrence of event types. Event type I has the highest number of events throughout the experiment. Right before the intersection of the growing fracture with the pre-existing fracture at 16097 seconds, there is a rapid increase in the number of events occurring from less than 5 to 21 events in less than 10 minutes.

3.2 Microseismic event locations

We are able to match the locations of the microseismic events with the key locations by examining plots of the microseismic event locations on still images from the video recordings of the experiments. From the proposed seismic model, we have 6 key locations with respect to the growing fracture (Table 3.2). The locations that we identified in our model are the expected source locations of the microseismic events. The key locations are 1) at the tip of the fracture, 2) associated with the tip but located within 1 cm of the fracture's edge, 3) within the body of the fracture, 4) ahead of the fracture with a distance > 1 cm from the visible fracture edge, 5) on the pre-existing fracture and 6) near the wellbore. There are events that occur ahead of the fracture but within 1 cm distance away from the fracture edge that we consider to be associated with the fracture edge.

For experiment 1, all the events occur in the area surrounding the wellbore and within the body of the fracture (Figure 3.8). We know that the locations of the events are within the body of the fracture and not at the tip from our video recordings and since the microseismic events occur ~ 40 minutes after the initiation of the fracture and the fracture dimensions pass the area near the wellbore (Appendix H.1).

One reason why the microseismic events in experiment 1 are located close to the wellbore (Figure 3.8) is because of the effect of the induced differential stresses near the wellbore on the growing fracture that produces microseismic events. Another possible but less likely cause of the microseismic events occurring close to the wellbore, is the creation of another fracture at the wellbore. From our images and from looking at the sample itself, we do not observe the formation of any other fractures. In our proposed mechanisms for the generation of microseismic events, we consider how induced stresses near the wellbore can

initiate twisting of the growing fracture and lead to the formation of mode III shearing features within the fracture itself (location 5 in Table 3.2). The shearing produced can be the source of the microseismic events. Further work with moment tensor analysis is necessary, however, to confirm the mode III shearing mechanism as the source of the microseismic events occurring close to the wellbore.

Table 3.2. Variations in the in the location of the events based on the event type for experiment 2

KEY LOCATION	NUMBER OF EVENTS	EVENT TYPES			
		I	II	III	IV
TIP	16	7	5	1	3
TIP AND AHEAD OF FRACTURE	8	2	3	3	0
ON/NEAR FAULT	56	29	10	11	6
NEAR WELLBORE	5	1	4	0	0
AHEAD OF FRACTURE	22	8	4	9	1
BODY OF FRACTURE	3	1	0	1	1

Experiment 2 has microseismic events located at each of the 6 key locations identified in our model (Table 3.2 & Figures 3.10-3.15). Most of the events in experiment 2 occur in relation to the pre-existing fracture or the tip of the growing fracture with 51% of the events occurring on or within 1 cm of the pre-existing fracture, 20% occurring ahead of the fracture, and 14.5% occurring at the fracture edge (Table 3.2). We observe that the microseismic event locations are mainly on the side of the fracture that is growing towards the pre-existing fracture (to the west of the well in Figures 3.10-3.15 & 3.18).

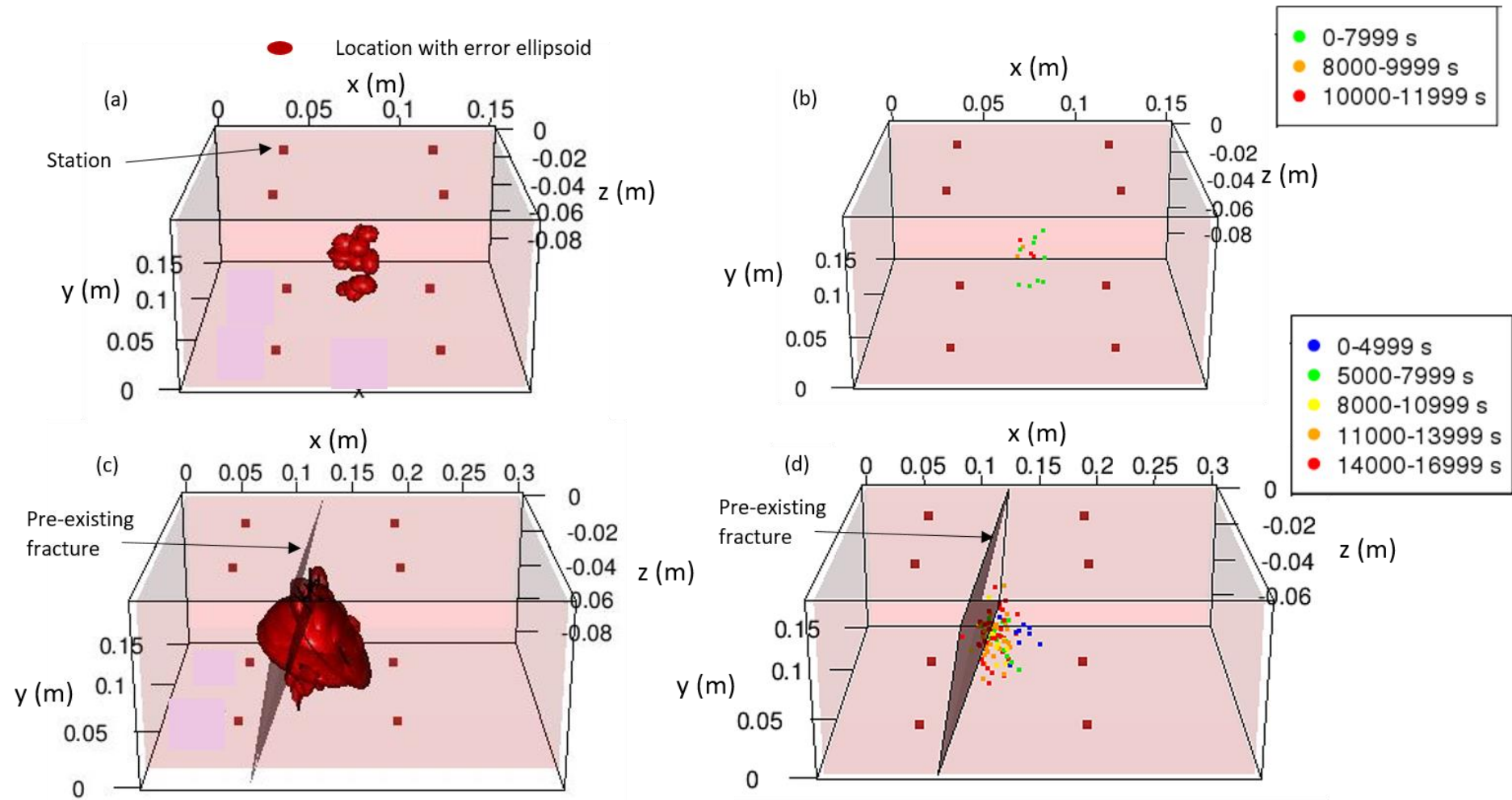


Figure 3.8. Locations of the events detected for experiments 1 & 2 with error ellipses, without the error ellipses and color coded based on the time of the occurrence of the microseismic event. (a) & (b) For experiment 1, most of the events occur in the vicinity of the well bore or within the body of the fracture close to the wellbore. (c) & (d) For experiment 2, the events identified occur at different locations throughout the fracture, near the wellbore and also in close association to the pre-existing fracture. (Appendix for 2D plots of locations)

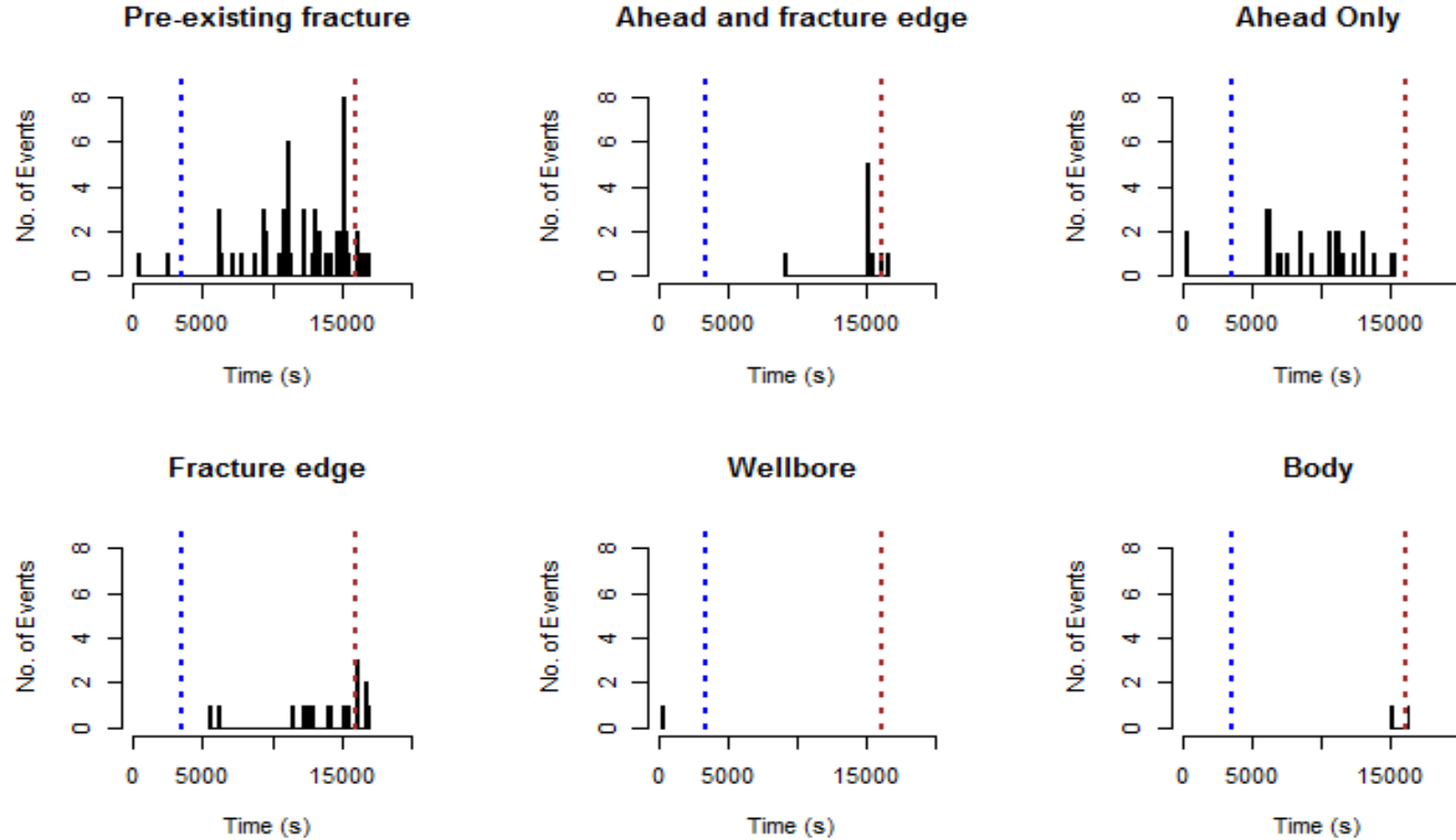


Figure 3.9. The locations of the events from experiment 2 in relation to the timing of their occurrence (Locations of the microseismic events in experiment 2 in Appendix F). The majority of events occur in relation to the fault (a) and ahead of the fracture (c). There is a general increase in the number of events as the fracture approaches the fault and intersects it (between 16,000 and 17,000 seconds). The blue dotted line is at fracture initiation and the brown dotted line is at the intersection of the growing fracture with the pre-existing fracture.

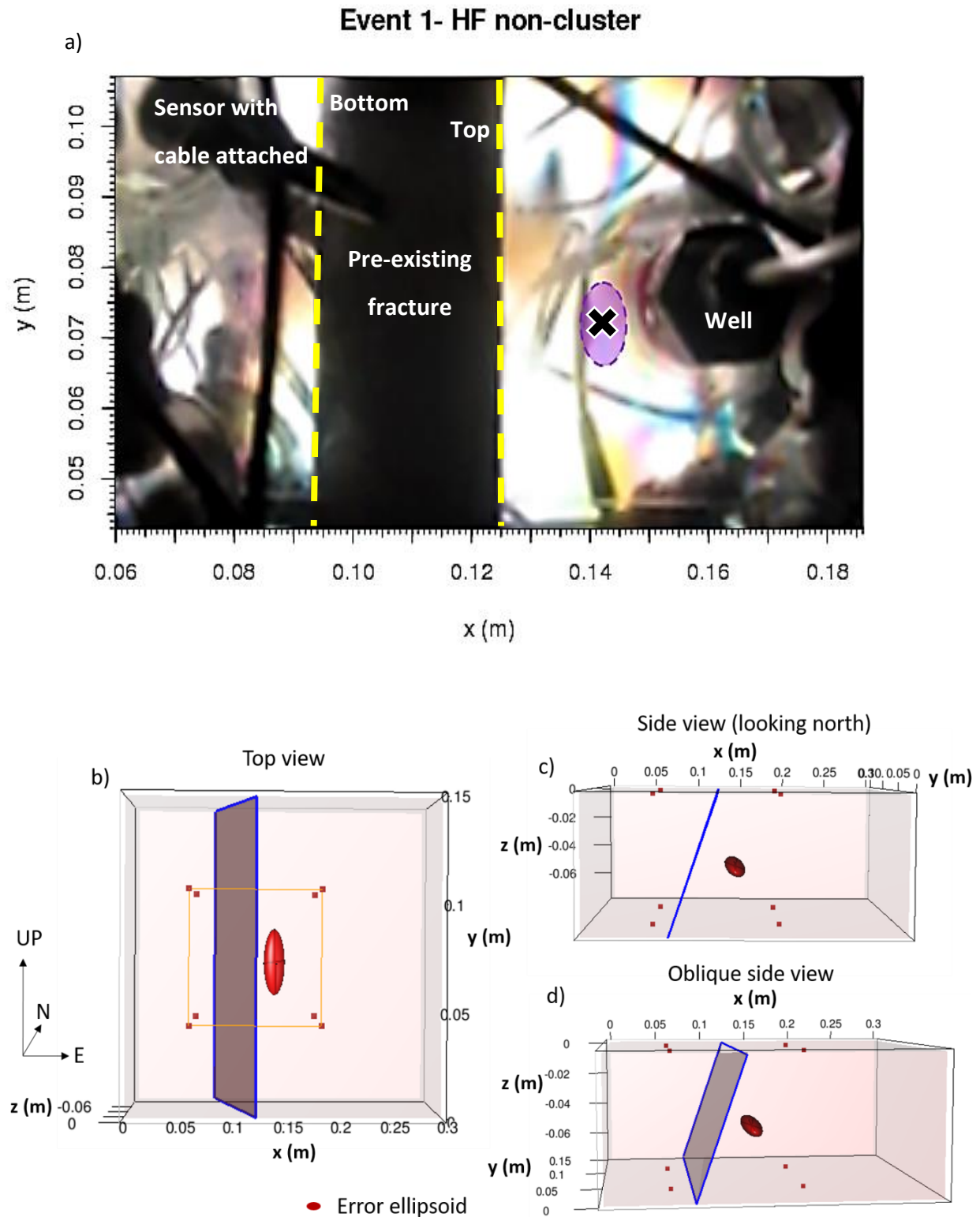


Figure 3.10 a) The location of Event 1 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline.

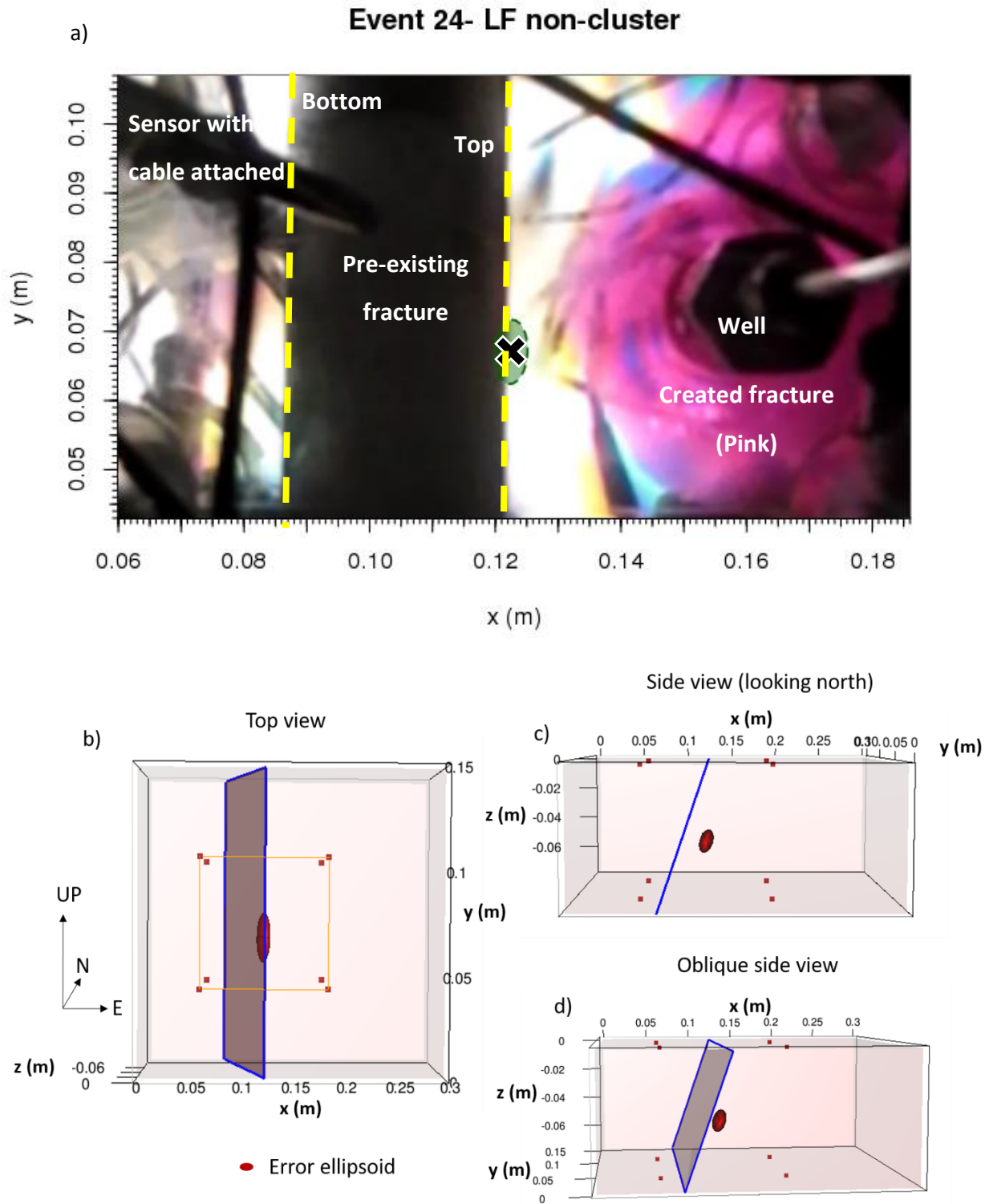


Figure 3.11 a) The location of Event 24 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 24 occurs ahead of the fracture and before the pre-existing fracture.

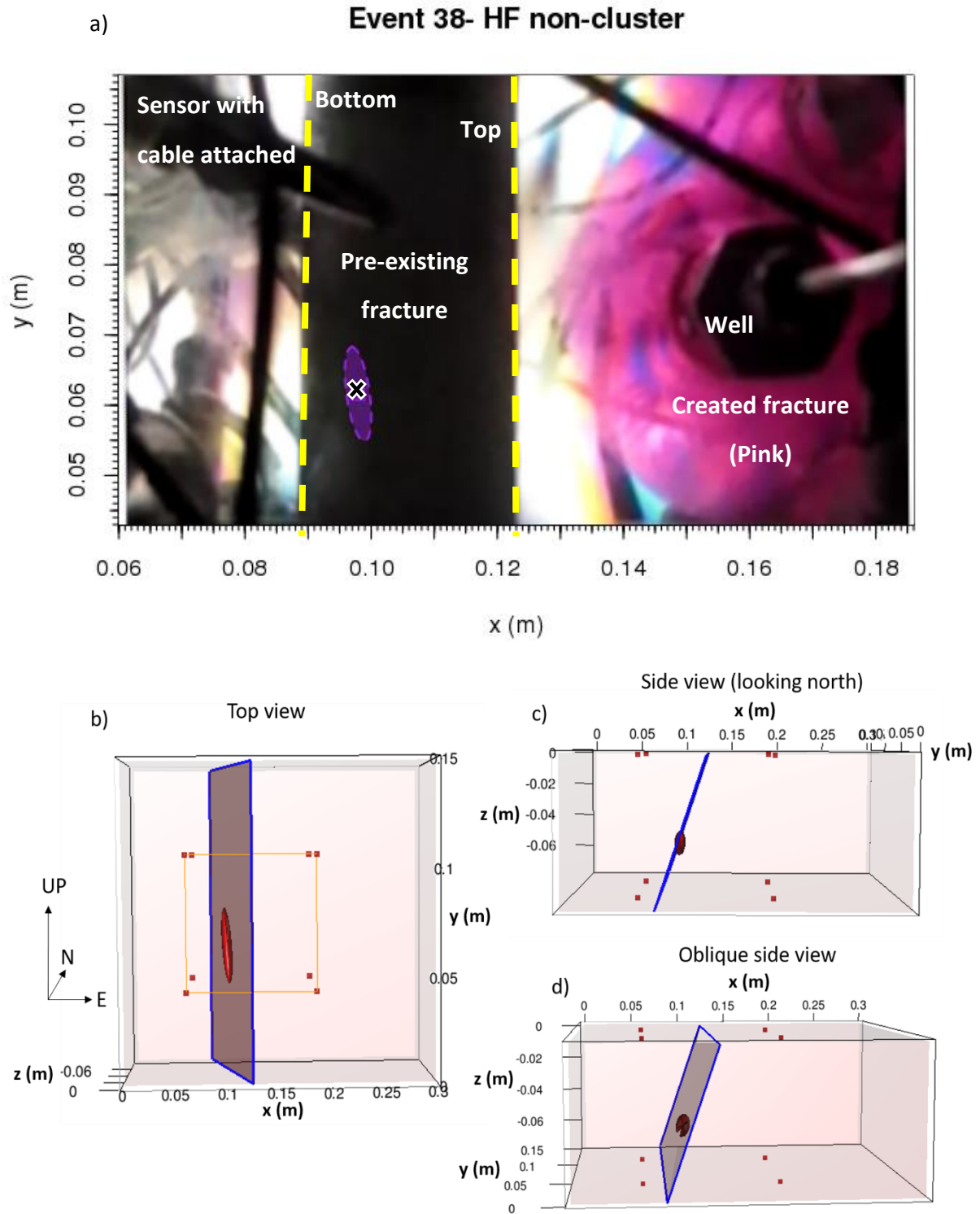


Figure 3.12. a) The location of Event 38 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 38 occurs on the pre-existing fracture as the fracture is still growing and has not yet intersected the pre-existing fracture.

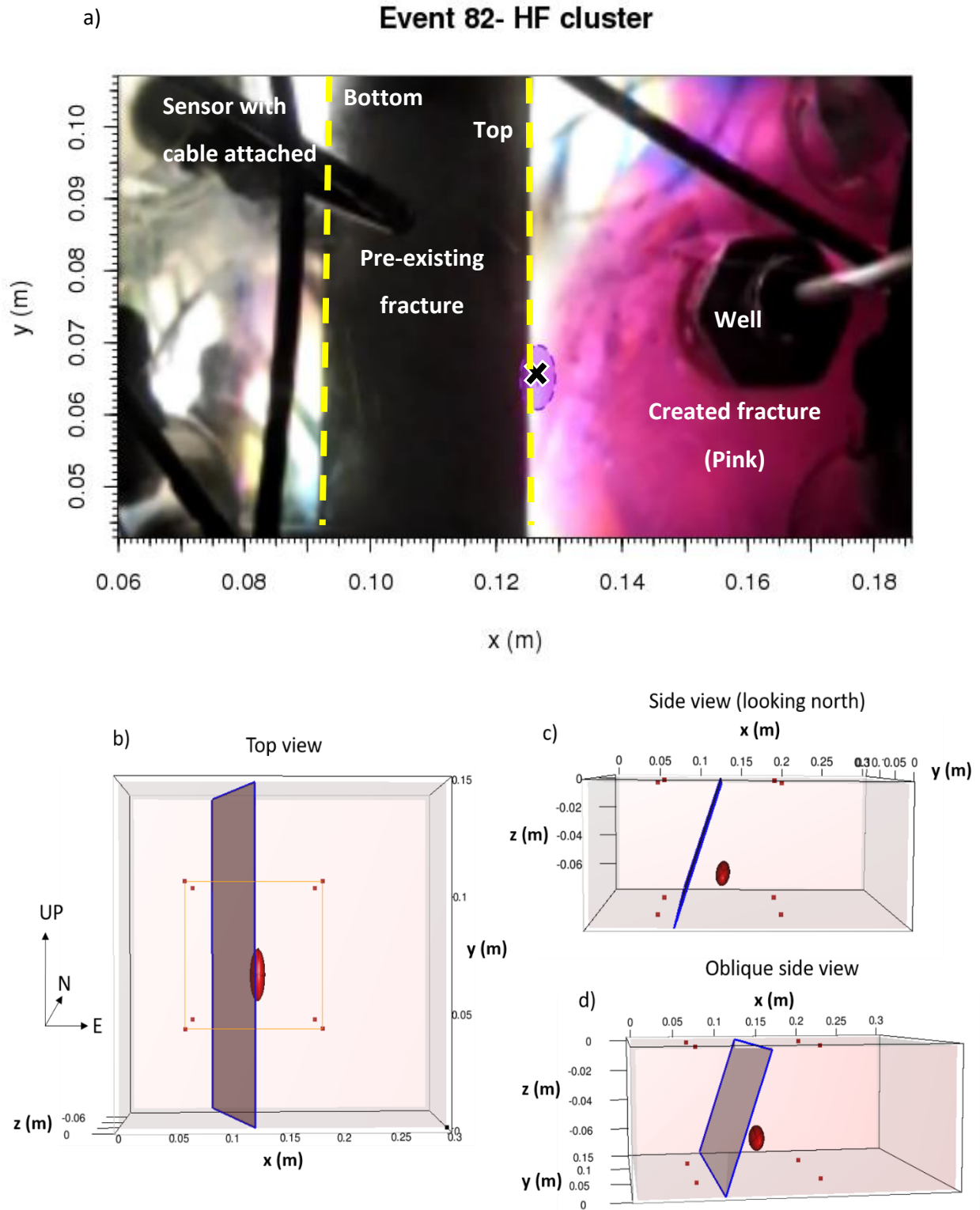


Figure 3.13. a) The location of Event 82 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 82 occurs at the tip of the fracture.

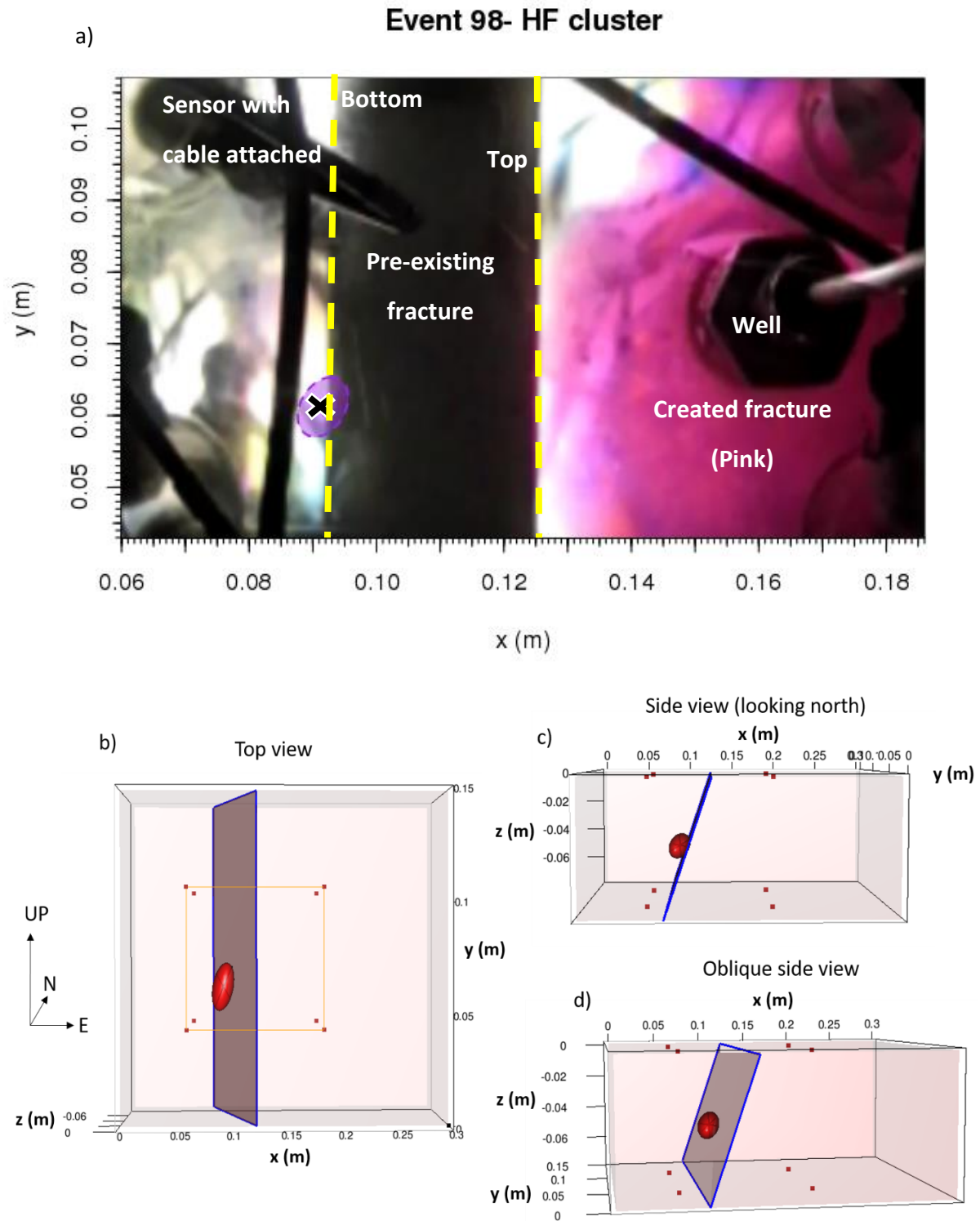


Figure 3.14. a) The location of Event 98 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 98 occurs on the other side of the pre-existing fault.

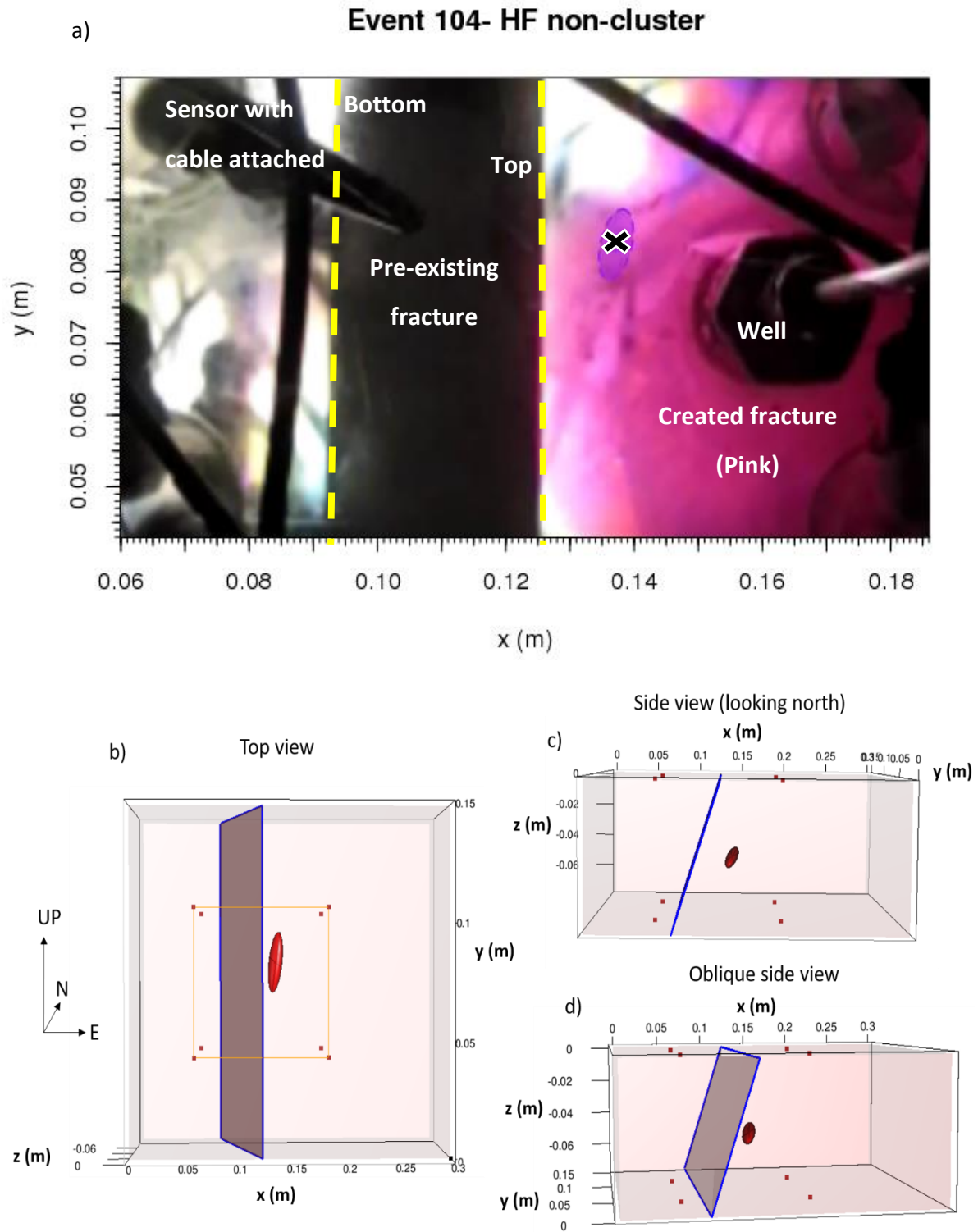


Figure 3.15. a) The location of Event 1 from experiment 2 shown on the snapshot from the video recording of the experiment at the time that the event occurred. b) Top down view of a schematic 3D diagram of the block. The orange box represents the field of view of the camera in the snapshot. c) & d) The side views of event 1 from experiment 2 in the schematic diagram of the block. The red cubes represent the location of the sensors, the pre-existing fracture is the grey plane with a blue outline. Event 104 occurs within the body of the fracture.

In experiment 2, there is a general increase in the number of events that occur on the fault and at the fracture tip over the course of the experiment. Right before the intersection of the growing fracture with the pre-existing fracture the highest number of microseismic events occur, with most of these events occurring as event type I and type II, located along the pre-existing fracture and on the fracture edge (Figures 3.7 & 3.9). The low frequency event types III and IV occur mainly in association with the pre-existing fracture and ahead of the growing fracture (Figure 3.7).

The relative lack of seismicity in experiment 1 suggests that hydraulic fracture growth is predominantly aseismic. The presence of the pre-existing fracture increases the degree of microseismicity that occurs as shown in the greater number of microseismic events occurring in experiment 2. In experiment 2, the general increase in the microseismic events throughout the experiment occurs because of the increased interaction of growing fracture with the pre-existing fracture as the distance between the two fractures diminishes (Figure 3.7).

The increase in the number of microseismic events, occurring along the pre-existing fracture and at the fracture tip, supports our model predictions of debonding and slip along the pre-existing fracture before the actual intersection of the two fractures. Specifically, we observe an increase in the number of high frequency event types I and II. The increase in the high frequency microseismic events is in accordance with our model where high frequency events occur along the fault prior to and at the intersection of the fault and the fracture. In our model, low frequency events and low frequency tremor (cluster of low frequency events) occur along a pre-existing fracture due to slow slip induced by the fracture approaching it.

The events occurring ahead of the visible edge of the fracture are likely to occur because of the differential rates of growth at different points along the fracture edge. There

is a lag in the growth of the fracture edge at the slower points of growth causing the edge of the fracture to not grow evenly. As the edge of the fracture becomes more even, the fluid fills up the areas to which the fracture extends. Along the edges of the fracture that extend, shearing may occur to produce the high frequency events observed ahead of the fracture edge. The low frequency events occur as the fluid flows into the newly created parts of the fracture.

3.3 Fracture dimensions

3.3.1 Fracture Shape

Experiment 1 produces an approximately circular or radial fracture, where length 1 is approximately equal to length 2 (Figure 2.12 & Figure 3.16). By plotting the two lengths on a graph, we can obtain the slope of the graph which is equivalent to the ratio of the two lengths. From our plot of length 1 and length 2 in experiment 1, we observe that the slope of the graph gives us a slope of approximately 1 and therefore there is a close 1:1 ratio for the two lengths (Figure 3.17). When we compare the shapes of the fracture from experiment 1 to the fracture in experiment 2, we can observe that in experiment 2, the fracture is more elongated in an easterly to westerly direction, while the fracture in experiment 1 is rounder and closer in shape to a circle.

In experiment 2, however, the shape of the fracture created is elliptical with more elongation on the side of the fracture closest to the fault (Figure 3.3). The growth of the fracture in experiment 2 is not radial as the slope of the graphs of length 1 and length 2 is 0.6 and not ~ 1 as in the first experiment (Figure 3.4). We note, however, that in experiment 2, the orientation change of the camera during the experiment does not allow for full measurement of length 2. We estimate of the length of the segment cut off in the images

from the video recordings after the orientation change and add this value to the portion of length 2 we are able to measure.

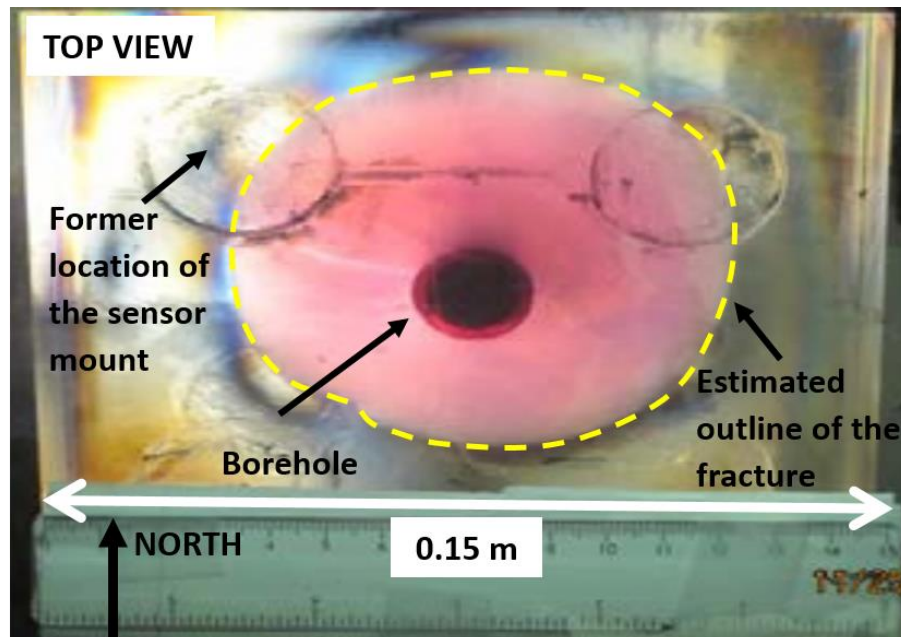


Figure 3.16. Top view sample 1 showing the fracture created in experiment 1. The pink fluid shows the extent of the fracture. The edge of the fracture itself is not fully rounded and smooth.

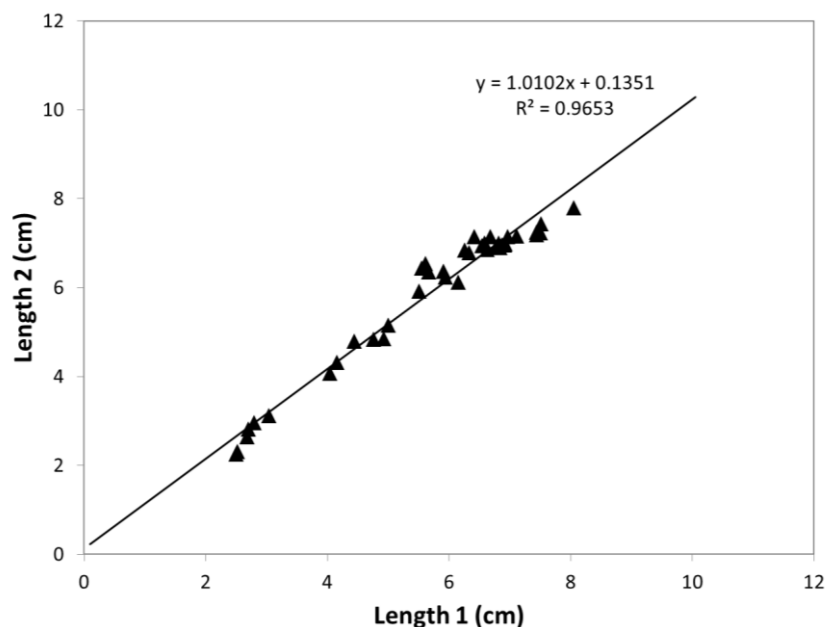


Figure 3.17. The relationship between length 1 and length 2 (as defined in Figure 2.11) of the created fracture in experiment 1. The slope of the graph shows a 1:1 relationship between the lengths verifying that the fracture grows radially throughout the course of the experiment.

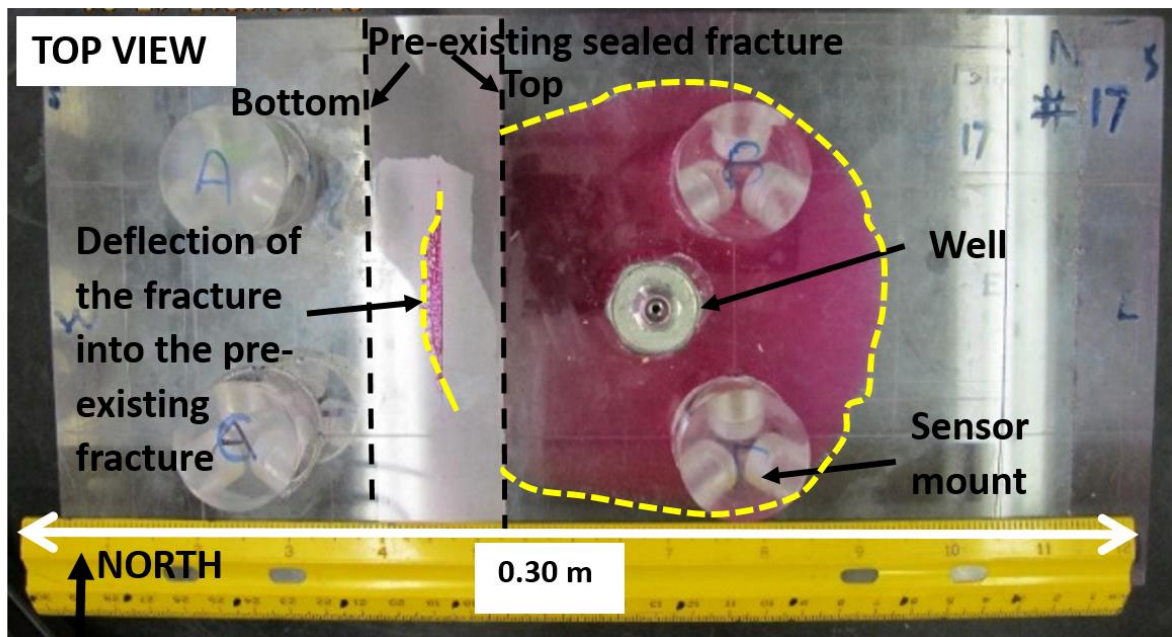


Figure 3.18. Top view of sample 2 showing the fracture created. The pink fluid shows the extent of the fracture. The created fracture intersects the pre-existing fracture (model fault) and is diverted into the fracture (pink semi-circle along the fault as labelled).

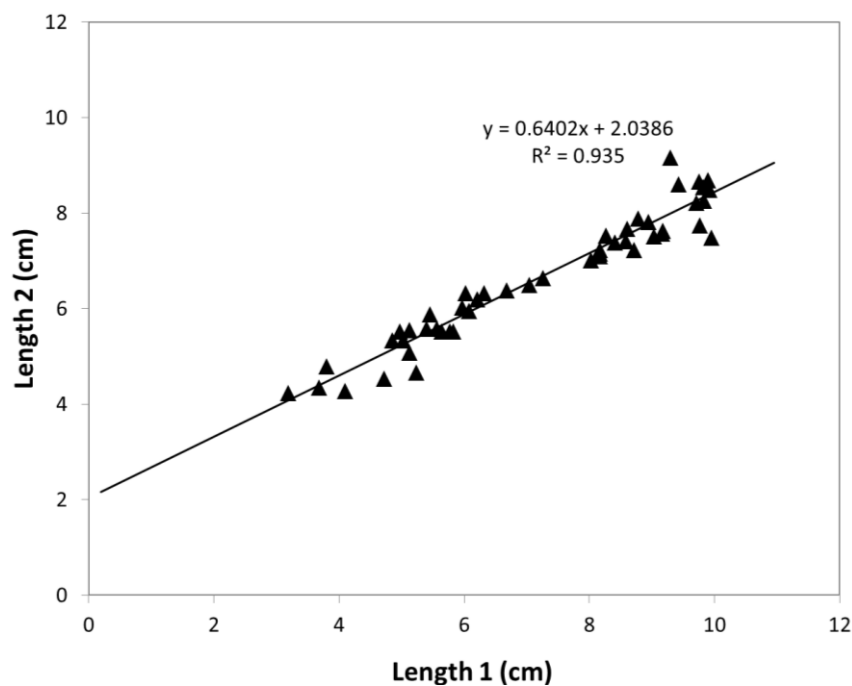


Figure 3.19. The relationship between length 1 and length 2 (as defined in Figure 2.12) for experiment 2. The slope of the graph shows that length 2 grows at a slower rate as compared to length 1 during the experiment. This indicates that during the experiment length 2 is smaller than length 1 suggesting that fracture growth is not radial in experiment 2.

The created fracture opens and propagates along the pre-existing fracture rather than cross into the opposite side of the block. The shape of the filled area of the pre-existing fracture is semi-circular with a length of 0.056 m measured in N-S direction along the pre-existing fracture, a length of ~ 0.01 m measured in E-W direction on the pre-existing fracture (Figure 3.18).

We observe that in experiment 2, the created fracture grows into the pre-existing fracture when the two fractures intersect. One way we can explain why the fracture is deflected downwards is to consider the energy needed for the probable paths of the fracture (figure 1.5). The fracture grows in the direction where energy release is greatest (Dahi-Taleghani and Olson, 2011). Therefore, the energy release is greatest for the fracture deflecting into the pre-existing fracture rather than the created fracture crossing the pre-existing fracture (Dahi-Taleghani and Olson, 2011).

The deflection occurs down in one direction because the fracture requires extra energy to be deflected in both directions along the fault and it is likely that debonding or opening along the pre-existing fracture is asymmetric with more of the pre-existing fracture opened below the fracture's plane of propagation (He and Hutchinson, 1989; Dahi Taleghani, 2009). Another reason why the fracture grows downwards is because of the force of gravity which forces the fluid to flow downward.

3.3.2 Fracture surface

The fractures created in both experiments in general grow horizontally. However, there are areas on the fractures that are not perfectly smooth (Figure 3.20). For example, there are concentric ramps on the fracture surface called ribs (Figure 3.16, 3.18 & 3.20). The

planes of the fractures in both experiment 1 and 2 are not horizontal and twisting of the fracture plane occurs because of the effect of tortuosity on fracture growth (Figure 3.20).

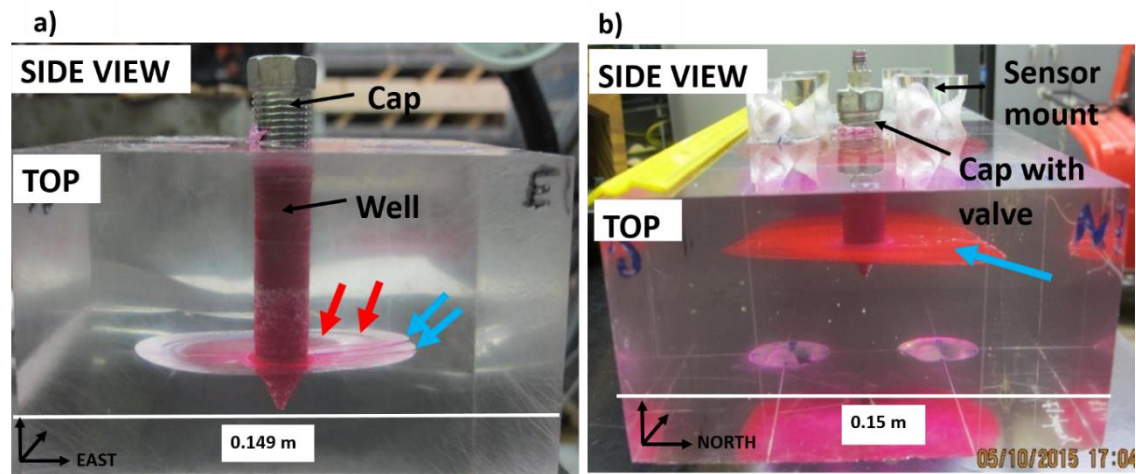


Figure 3.20. Side views of the sample from experiment 1 (a) and the sample from experiment 2 (b). In (a), there are two linear features that show vertical displacement on the fracture surface (blue arrows) and rib-marks (red arrows). In (b), we see irregularities on the surface of the fracture as well (blue arrows). The fractures in both samples show some evidence of twisting in the fracture growth rather a perfectly horizontal fracture surface.

As the fracture twists out of the original plane of propagation, mode III shear fractures and segmentation on the fracture occurs (Figure 3.20). The most likely interpretation for the cause of the high frequency microseismic events in experiment 1, occurring near to the wellbore is the mode III shearing.

3.3.3 Fracture dimensions

The rate of growth of the fractures is different in the 2 experiments. For each measurement, that is length 1, length 2, perimeter and area, we see similar changes in the slopes at the same times (Figure 3.21). We use length 1 as an example of these changes in the slopes of the graphs which represents the rate of growth of the fracture.

For experiment 1, there are 4 distinct slope variations on the graphs of the fracture dimensions (Figure 3.21). For the first period of fracture growth, between 6,000 seconds and

6,500 seconds, there is a gradual increase in the size of the fracture as we observe in the rate of growth of length 1 of 3.084×10^{-4} cm/s. The second period, from 6,500 seconds to 8,500 seconds, of rapid growth of the fracture, where the rate of growth of length 1 increases to 1.75×10^{-3} cm/s. This period of rapid growth precedes a slower growth rate, from 8,500 seconds to 10,500 seconds where the growth rate of length 1 decreases to 2.5×10^{-4} cm/s. From 10,500 seconds and to the end of the experiment, the fracture rapidly increases in size, with length 1 increasing by a rate of 9.33×10^{-4} cm/s (Figure 3.21).

For experiment 2, there are 3 slope variations on the graphs of the fracture dimensions (Figure 3.22). There is rapid growth of the fracture for the first period from 3,000 seconds to 5,000 seconds corresponding to the initiation of fracture growth. The rate of growth of length 1 is 9×10^{-4} cm/s for the first period. The growth of the fracture for the second period of growth from 5,000 seconds to 15,000 seconds is slower than that of the first period with a rate of 4.356×10^{-4} cm/s, approximately half of the rate of growth of the first period. The third period of growth from 15,000 seconds until the end of fracture growth, shows very little change in the size of the fracture with the rate of growth for this period $\sim 1 \times 10^{-5}$ cm/s. During this period, the fracture is approaching and intersecting the pre-existing fracture, the fracture growth becomes constant as the growing fracture is diverted into the pre-existing fracture.

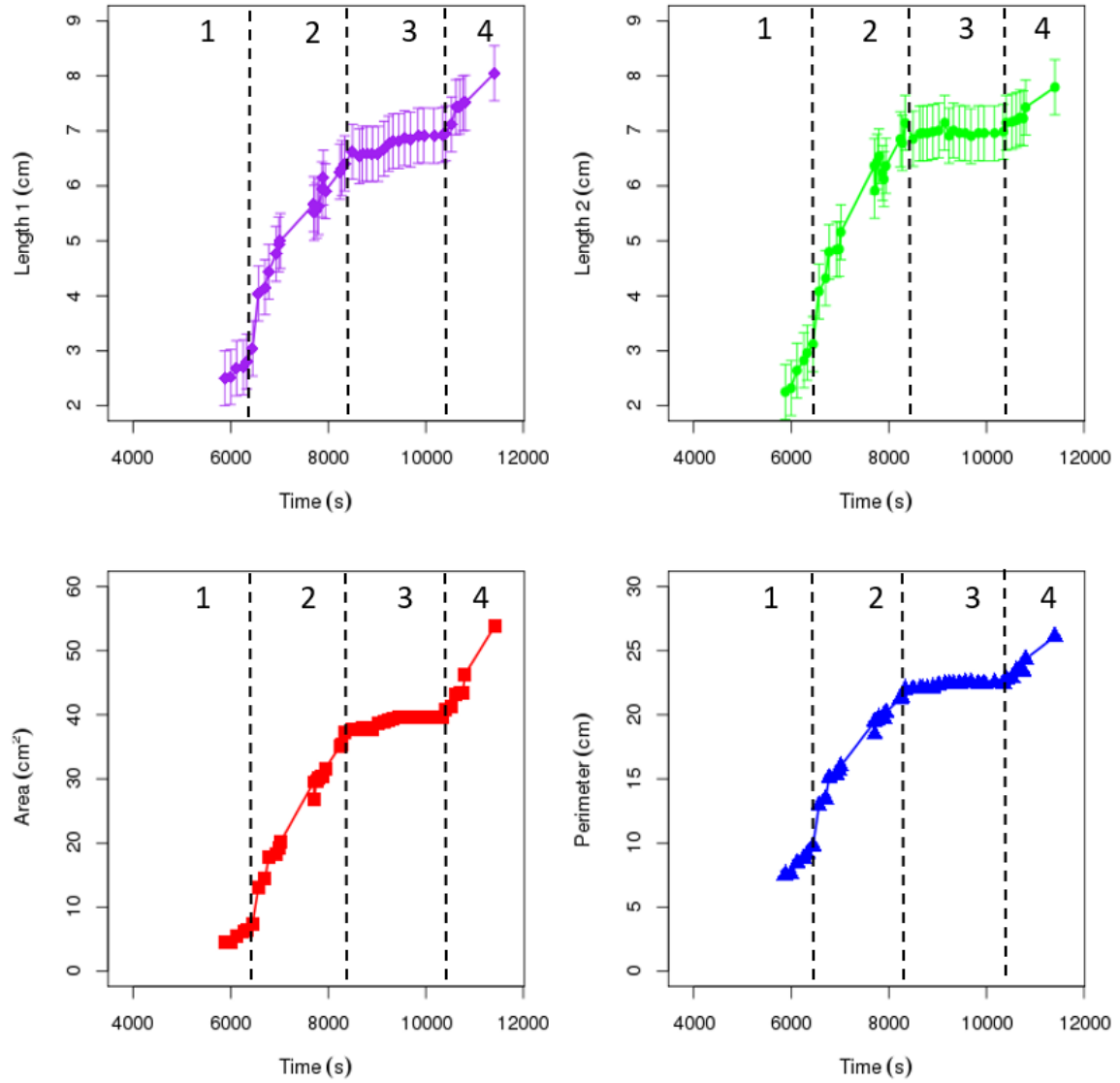


Figure 3.21. For experiment 1, fracture growth measurements, Length 1 and Length 2 (as identified in methods showing the measurement of the fractures using ImageJ software), the area, and the perimeter of fractures. The error in measurement for the values including human error in measurement and ImageJ errors is ± 0.05 cm.

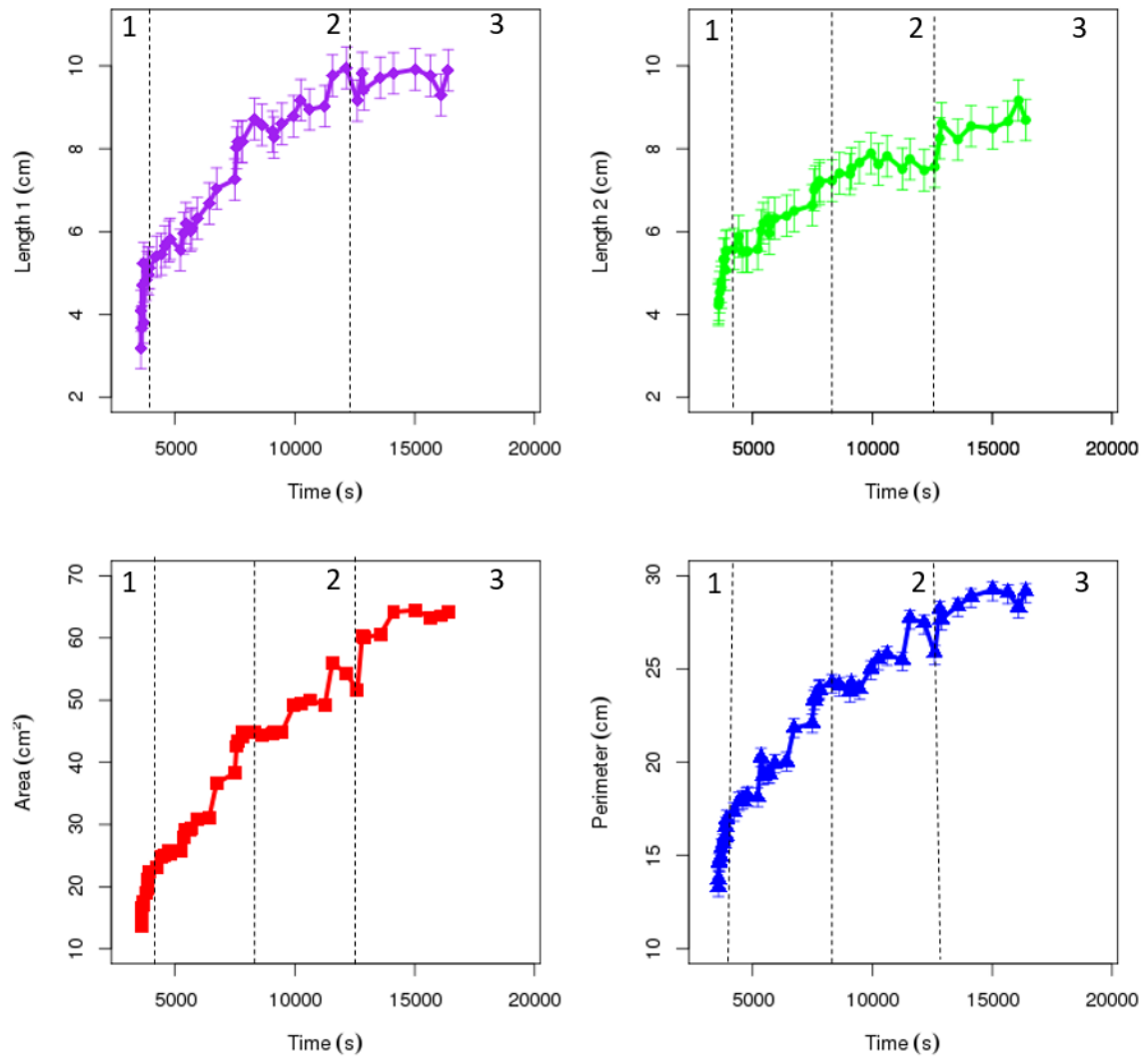


Figure 3.22. For experiment 2, fracture growth measurements, Length 1 and Length 2 (as identified in methods showing the measurement of the fractures using ImageJ software), the area, and the perimeter of fractures. The error in measurement for the values from human error in measurement and ImageJ errors is ± 0.05 cm. The dashed lines shown on the plots represent time that the camera position changes and the crack is not fully seen in the recordings after this point. We add an estimate of the missing section of the crack to the measurements of the crack that we are able to obtain from the video recordings.

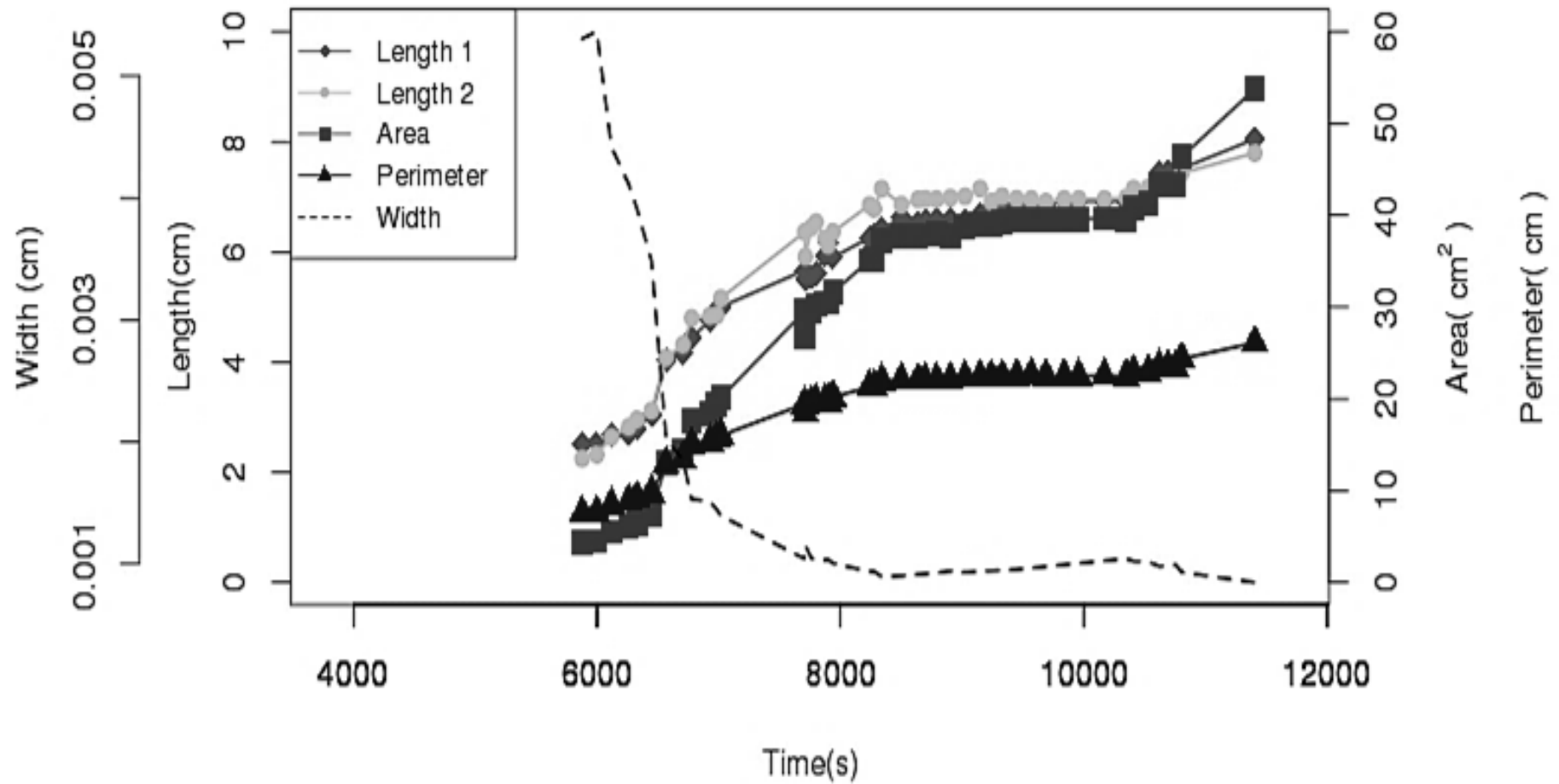


Figure 3.23. The lengths (1 and 2), area, perimeter, and average width of the created fracture during the experiment 1. As there is a rise in the lengths, area and perimeter of the fracture, there is a decrease in the width generally. There are two points where there is a small increase and then a rapid decrease in width size occurring at $\sim 8,000$ seconds and $10,500$ seconds.

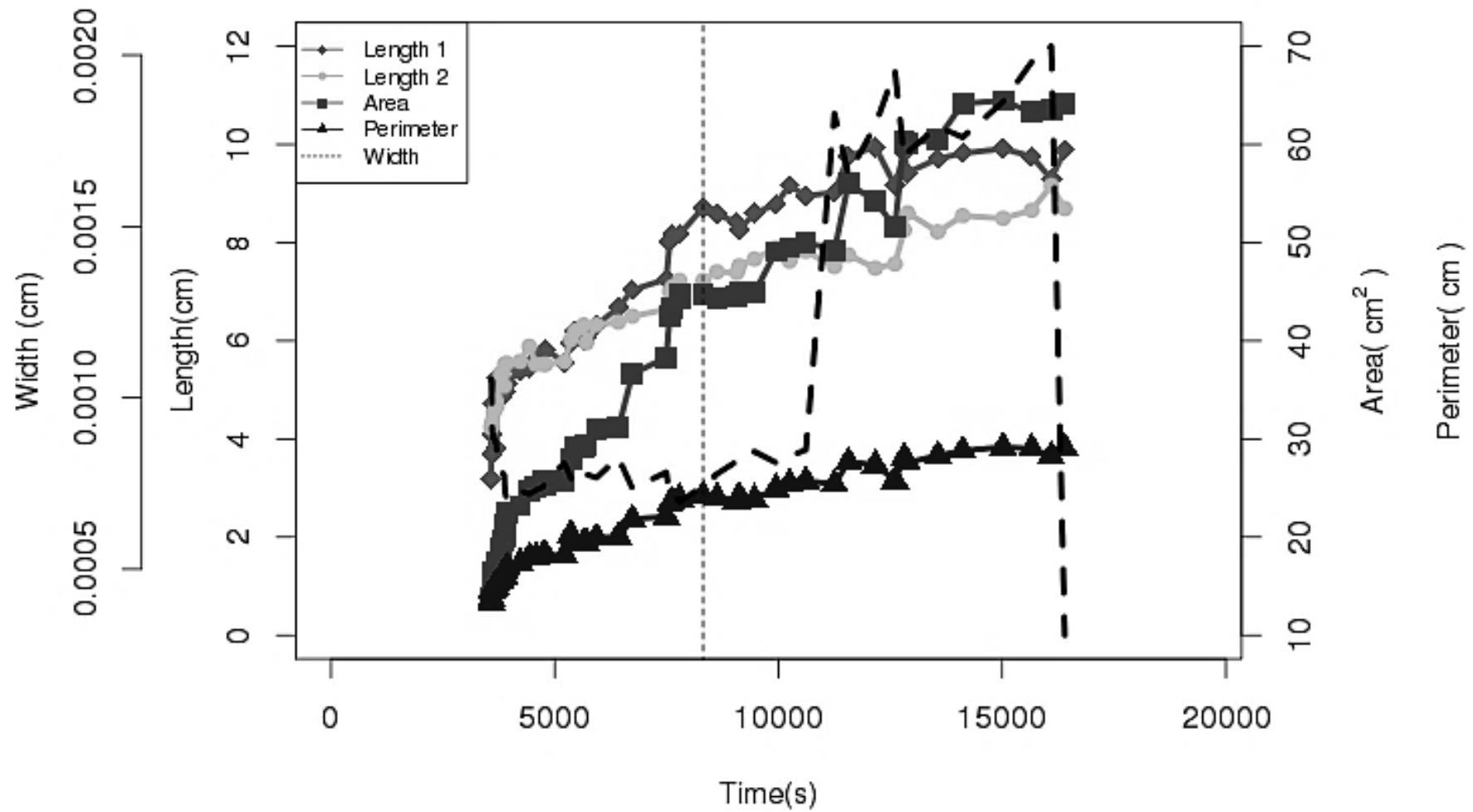


Figure 3.24. The lengths (1 and 2), area, perimeter, and average width of the created fracture during the experiment 1. As there is an increase in the lengths, area and perimeter of the fracture, there is also a general increase in the width size (Figure 3.18). There are fluctuations as well in the width size similar to fluctuations observed in the pump pressure response. The dashed line represents the time in the experiment where the camera changes orientation.

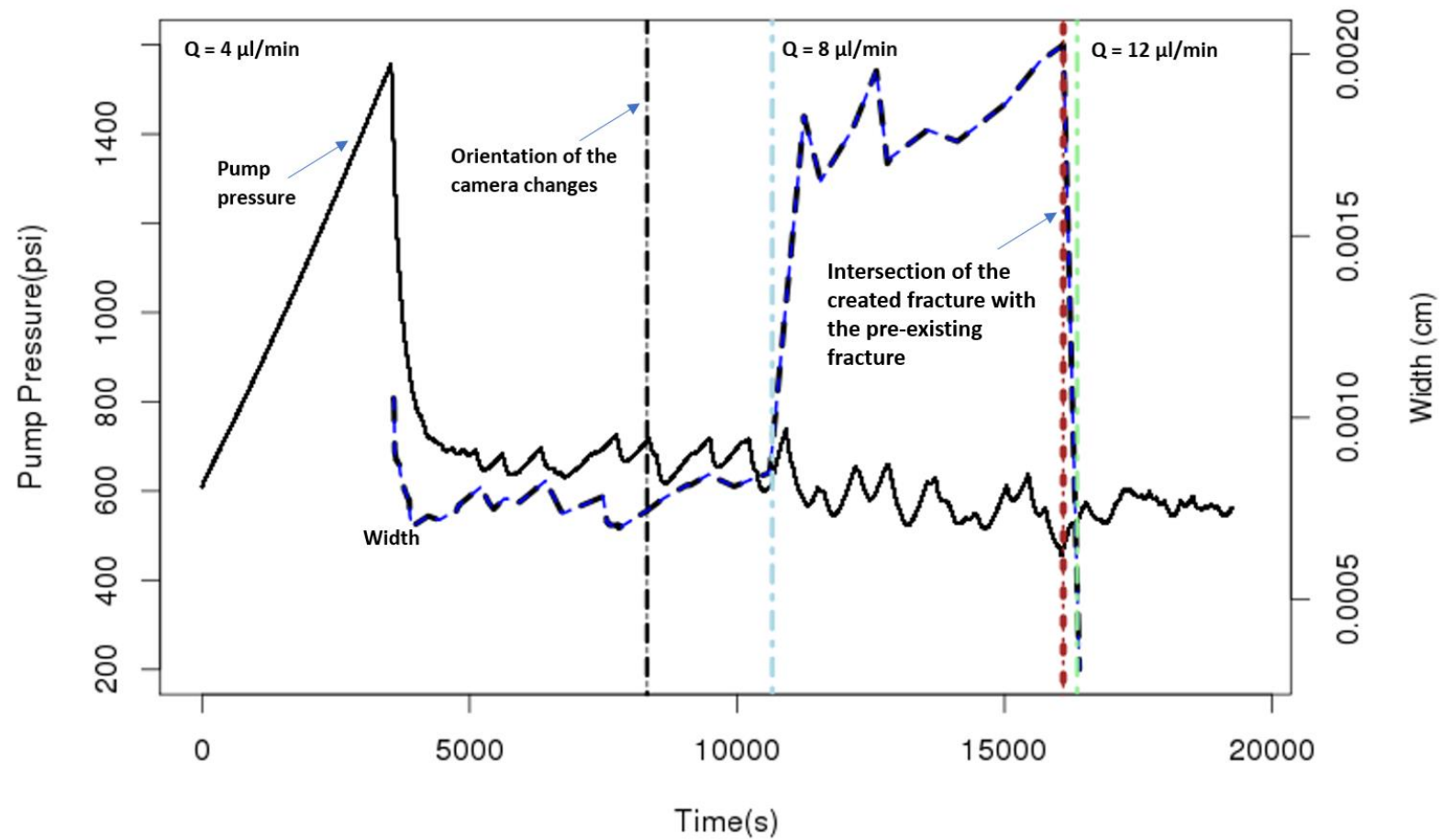


Figure 3.25. The variation of fracture width during experiment 2 and pump pressure response. The width initially decreases rapidly at the start of fracture growth similarly to the decrease in the pump pressure before 4,800 seconds after which an inverse relationship is observed where there is a decrease in the pressure response and there is a peak in the width size.

We directly measure the fracture dimensions in two-dimensions from the camera images. Since we know the volume of fluid pumped into the fracture during the experiments and the 2-dimensional measurements, we calculate the average width using the material balance of the volume pumped into the sample and the volume of the fracture. For the material balance equation,

$$\text{Volume pumped} = \text{Volume in the fracture} + \text{Volume lost}, \quad 3.1$$

Since the samples are relatively impermeable, the leak-off of the fracturing fluid into the sample (Volume lost) is deemed negligible for these calculations (Jones and Britt, 2009).

Therefore,

$$\text{Volume pumped} = \text{Volume in the fracture} \quad 3.2$$

$$Q \times t_p = w \times L_1 \times L_2 \quad 3.3$$

$$w = \frac{Q \times t_p}{L_1 \times L_2} \quad 3.4$$

where Q is the pump rate, t_p is the pump time, L_1 is length 1 of the fracture, L_2 is length 2 of the fracture and w is the average width of the fracture (Jones and Britt, 2009).

For experiment 1, the average width is largest at the initiation of the fracture and then generally decreases as the fracture propagates (Figure 3.23). The average width decreases to approximately 0 when the experiment stops. For experiment 2, there is an initial decrease in the size of the average width, followed by alternating periods of increasing and decreasing average width (Figure 3.24 & 3.25). There is a rapid increase in the average width at $\sim 10,800$ seconds again followed by a period of alternating increases and decreases in the average width.

When the average width of the fracture decreases, it can suggest two scenarios, one where the fracture has encountered a barrier or higher stress that forces it to stop propagating and thus close or it indicates that the fluid pressure within the fracture is not large enough to keep the fracture open so the fracture closes (Figure 3.25). Since pumping is not stopped, when the fracture closes, we observe an increase in the fluid pressure as the fluid pumped into the fracture is more than the available space. With the two scenarios, when the fracture is able to continue propagation by overcoming the barrier or the pressure increasing enough to reopen the fracture, the fluid flows into the new space created and the pressure drops. We observe the inverse relationship between the width and the pump pressure mostly in experiment 2, where as the average width increases, the pump pressure decreases and vice-versa (Figure 3.25).

3.4 Pressure response

3.4.1 Pressure Variations

Since the stresses applied to the block, the fluid viscosity and pump rate are constant throughout experiment 1, the treating pressure adjusts to maintain the balance as the fracture grows and L_1 and L_2 increase. For experiment 1, where the pump rate is constant at 4 $\mu\text{l}/\text{min}$ for the duration of the experiment, the increases in L_1 and L_2 determine how P_{net} varies throughout the experiment.

We observe an increase in the pump pressure response soon after pumping begins in both experiments (Figures 3.26 & 3.28). In experiment 1, the pressure gradually increases until the breakdown pressure is reached (~ 810 psi), where it levels and then rapidly declines until the pressure reaches approximately 450 psi (Figures 3.26 & 3.27). The pressure decline is gradual with two main drops in the pressure both ~ 50 psi. Overall the pressure declines to 350 psi at

the end of the experiment. The greatest number of seismic events occur where the rate of growth of the area of the crack is fastest i.e. the graph of the area of the crack has a steep slope (Figure 3.26 & 3.27). These events mainly occur at or along the rapid drops in pressure mentioned with only 2 of the 16 identified events occurring on the gentler slope of the decline.

In experiment 2, the pump pressure gradually increases until the breakdown pressure of ~ 1600 psi is reached. The pump pressure then rapidly declines until the pressure reaches approximately 650 psi (Figure 3.28). The overall pump pressure decline is gradual after the first main pressure drop, however there are many oscillations in the pressure response during the decline in the pressure. The oscillations in pressure have amplitudes that vary from approximately 50 psi to 100 psi (Figure 3.28 & 3.29).

From the video recordings of experiment 2, we note that there are periods of crack growth where the advancing front is not growing uniformly. The non-uniformity of the growth of the fracture may be related to the oscillations also observed in the average width of the fracture. From our calculations of the average width of the fracture in experiment 2, the average width values appear to follow the fluctuations in the pressure data. For most points in the measurements of the average width, where there is a peak in the pressure fluctuation, the width is decreasing and where the width is increasing, there is a decrease in pressure. At the When the growing fracture intersects the pre-existing fracture, there is a rapid decrease in the width of the growing fracture as fracture growth ceases (Figure 3.25). The intersection also corresponds to a decrease in pressure when the fluid enters and flows along the pre-existing fracture (Figure 3.28).

We can interpret the pressure responses in the two experiments by considering the changes in that occur in the samples at the start of pumping. The pressure is expected to increase since the fracture has not yet initiated and there is no space for the fluid being pumped into the samples to flow. As a result, the fluid experiences compression in the well which causes an increase in pressure (Kim and Wang, 2011). As the pump pressure increases so that the P_{net} is equal to the closure pressure, fracture growth initiates.

As the fluid pressure continues to increase, the fracture width increases rapidly as the fracture opens and the fluid from the well fills the volume created (Kim and Wang, 2011). The fracture also increases in size rapidly (Figures 3.23 & 3.24). When the breakdown pressure is reached, the pressure rapidly drops since there is less fluid to fill the space created by the growing fracture (Kim and Wang, 2011). The decline in the pressure in experiment 1 is a smooth curve which may suggest that the growth of the fracture becomes more stable after the rapid initiation of fracture growth (Figure 3.23).

The video recordings, pump pressure fluctuations and the fracture length and width measurements point to the possibility of different rates of growth at different points along the fracture edge. There may be periods of arrest and advancement at different locations along the fracture edge that can cause the fluctuations in the width size since as the fracture stops growing the width is expected to be smaller as the fracture closes and as the fracture grows the width is expected to be larger. Then, If the width closes, the fluid pressure within the fracture increases to open the fracture in order to continue the propagation of the fracture.

When the created fracture intersects the pre-existing fracture the created fracture grows into the pre-existing fracture and is not creating new pathway for the fracture to grow and

thus we expect the width to decrease. A probable reason for the drop in the fluid pressure at the intersection of the two fractures is the fluid has more space to flow as the growing fracture and the pre-existing fracture are now connected and opened after the intersection thus requiring less pressure for the continued flow of the fluid.

3.5 Relating the pressure response, seismic event occurrence and fracture growth

The seismic events occur during periods where the pressure decreases and fracture growth is relatively rapid. 14 out of the 16 seismic events in experiment 1 occur at or along the rapid drops in pressure with 10 events occurring between 7,800 – 8,000 seconds and 4 occurring between 10,900 – 11,000 seconds (Figure 3.26). The event with the highest maximum amplitude (1,449 counts) for experiment 1 occurs at ~ 7900 seconds (near the end of the interval of rapid pressure decline). The event is located close to the wellbore, as an event type II-b (high frequency cluster with more than two sub-events). During the time periods of the pressure drops, fracture growth is rapid (2 in Figure 3.21) compared to other periods during the experiment (Figure 3.26).

In experiment 2, there are fluctuations in the number of seismic events but the number of events do not appear to follow the oscillations in the pressure data (Figure 3.29). The event with the highest maximum amplitude (32,552 counts) occurs at ~ 300 seconds, on the pre-existing fracture, as an event type IV-b (low frequency cluster with more than two sub-events). There is no clear relationship between the fracture growth and seismic event occurrence.

We observe an overall increase in the number of seismic events as the growing fracturing approaches the pre-existing fracture (Figure 3.28). The largest number of seismic events occur at 15,000 seconds, right before (~1,000 seconds before) the intersection of the

two fractures and a decrease in the pump pressure. The increase in the seismic events as the growing fracture gets closer to the pre-existing fracture, strongly suggests that there is increased interaction between the two fractures. Furthermore, from the locations of the seismic events (Figure 3.9), we observe that most of the events occurring at ~15,000 seconds occur on the pre-existing fracture suggesting that slip may be occurring along the pre-existing fracture before the intersection. The evidence of the increase seismicity and the locations of the events support the ideas that the stress fields of the growing fracture and the pre-existing fracture interact long before the two fractures physically touch (Dahi Taleghani, 2009; Aimene and Nairn, 2014).

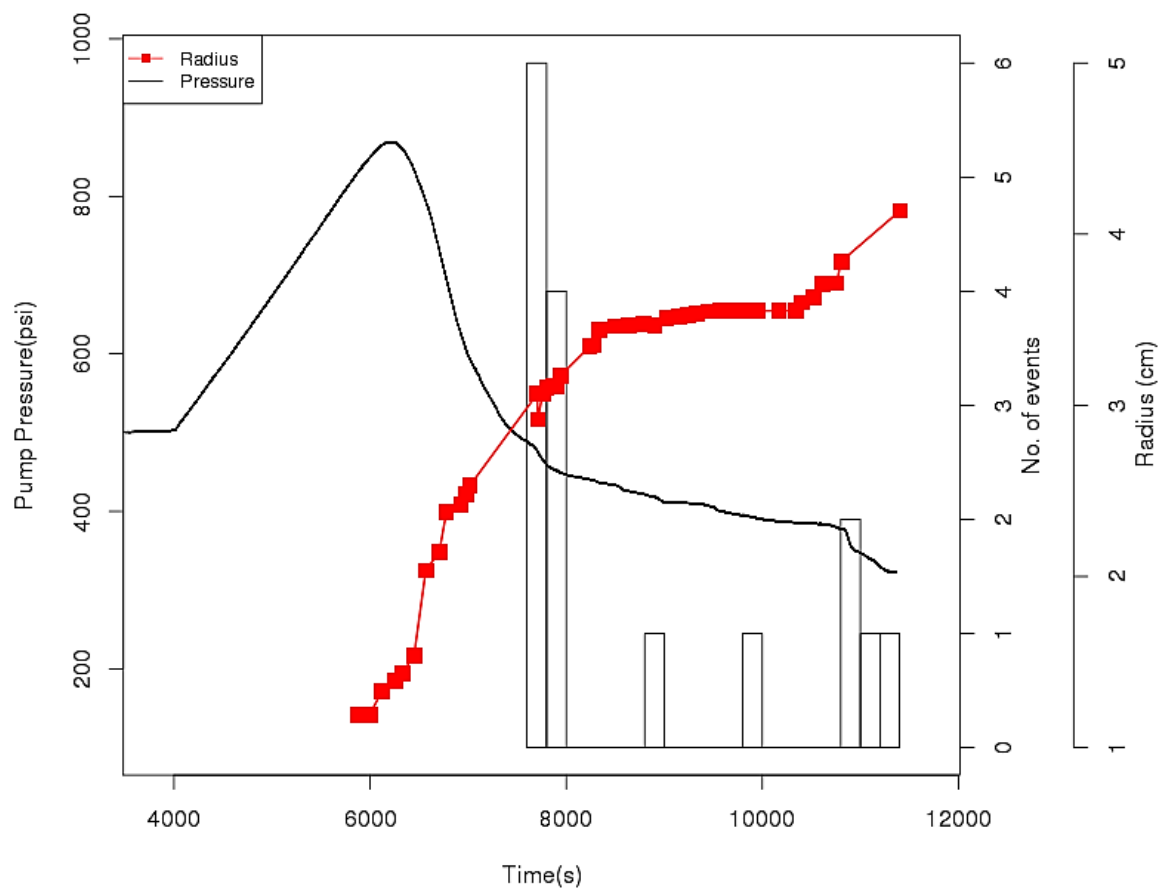


Figure 3.26. The number of seismic events for 5 minute intervals in relation to the rate of growth of the crack's radius calculated from the area ($r = \sqrt{(A/\pi)}$) and the pump pressure response for experiment 1. The error in measurements are ± 0.9 cm for the area.

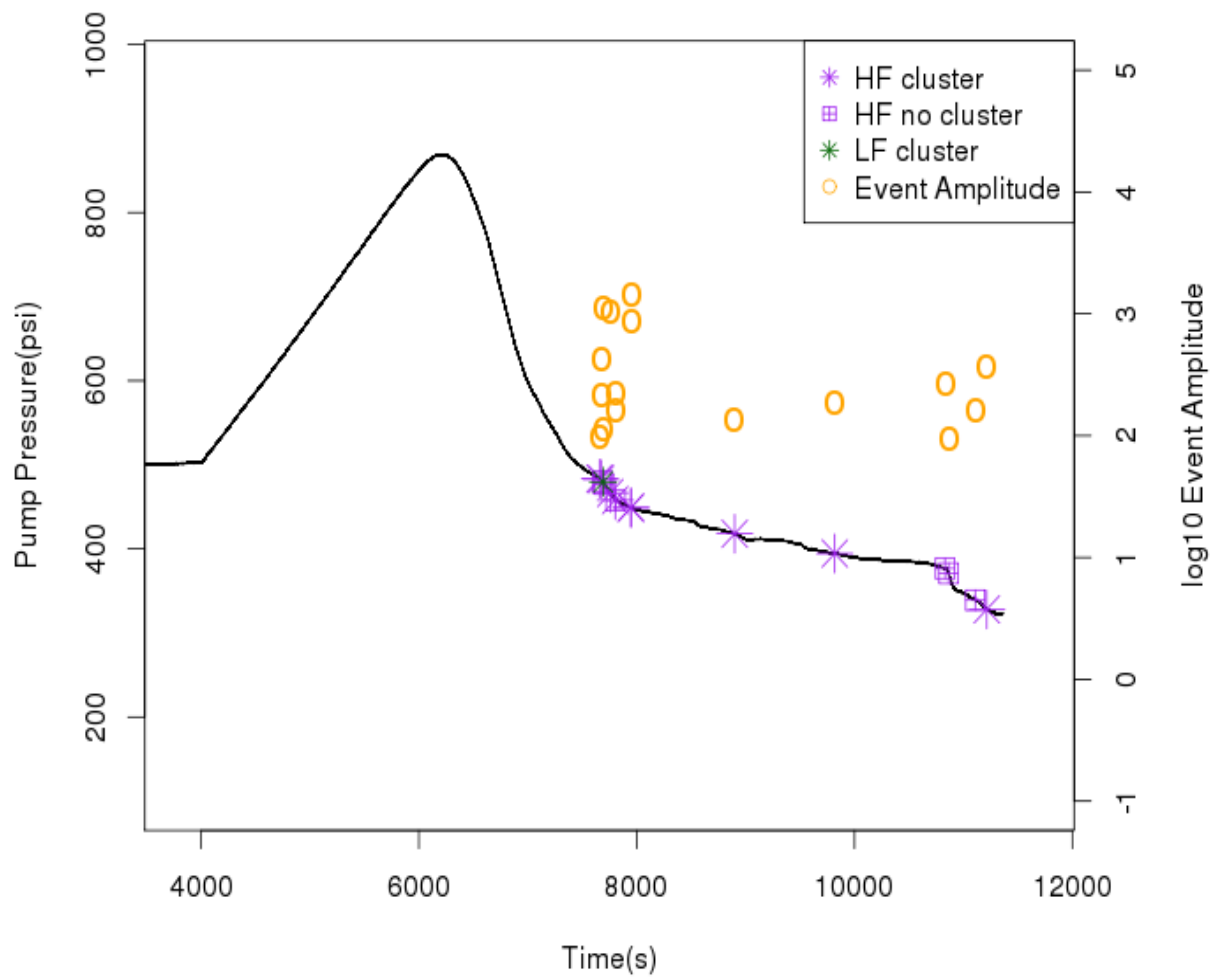


Figure 3.27. Pump pressure data, microseismic event occurrence with respect to time and event amplitude for experiment 1. The events with the greatest amplitudes correspond to the timing of the drops in the pressure response. The microseismic event with the greatest maximum amplitude (amplitude is on a log scale) occurs right after the first pressure drop near the onset of fracture growth.

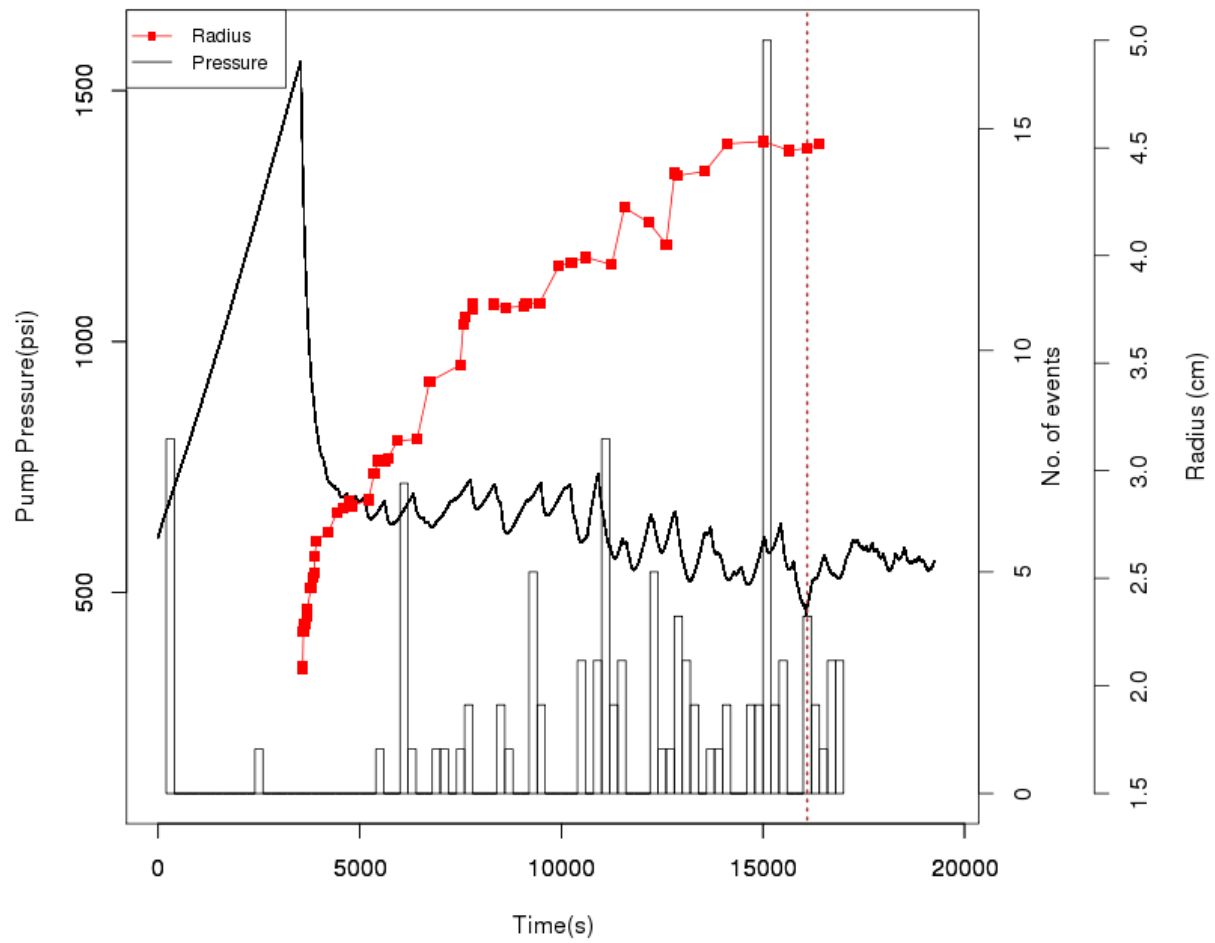


Figure 3.28. The number of seismic events in 5 minute intervals, the rate of growth of the crack's radius calculated from the area ($r = \sqrt{(A/\pi)}$) and the pump pressure for experiment 2. The error in measurements are ± 0.9 cm for the area. The dotted brown line indicates the intersection of the growing fracture with the pre-existing fracture.

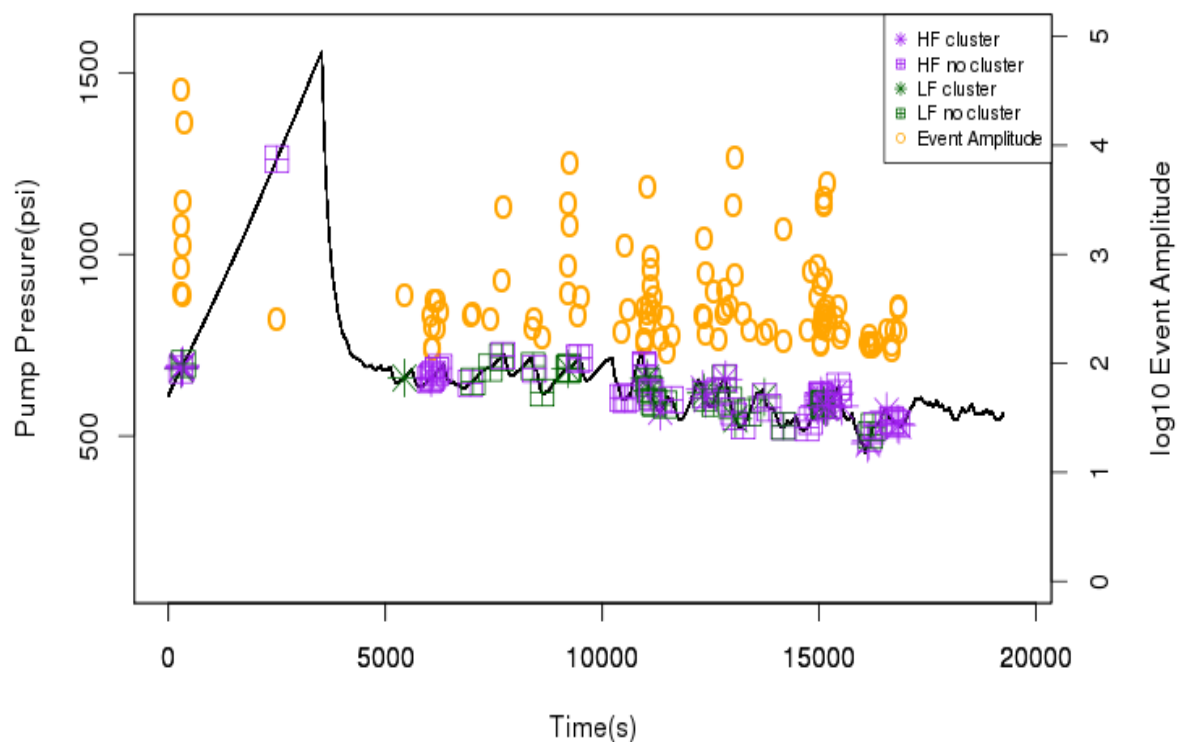


Figure 3.29. Pump pressure data, microseismic event occurrence with respect to time and event amplitude for experiment 2. The events with the greatest amplitudes do not necessarily correspond to the timing of the drops in the pressure response as observed in experiment 1. The microseismic event with the greatest maximum amplitude (amplitude is on log scale) occurs right before the first pressure drop and before the onset of fracture growth. The events observed before fracture growth when located are related to the pre-existing fault and the borehole with the largest amplitude event before the pressure drop located on the fault.

Chapter 4 Discussion

4.1 Comparing laboratory seismic data to seismic data in the field

We scale our laboratory seismic data and field seismic data in order to be able to use the seismic events produced in the laboratory to better understand the evolution of the seismic events produced during hydraulic fracturing in the field. Previous laboratory studies on magmatic fracturing use the ratio of the dominant frequency and the length of the fracture to scale the laboratory results up to the results expected in the field (Burlini et al., 2007; Benson et al., 2008) such that,

$$d_1 \times f_1 = d_2 \times f_2, \quad 4.1$$

and then

$$\frac{d_2}{d_1} = \frac{f_1}{f_2} \quad 4.2$$

where d is the length of the fracture and f is the dominant frequency of the seismic event produced in the field (1) and in the laboratory (2) respectively (Burlini et al., 2007; Benson et al., 2008). The ratio of the frequency and the size of the fracture producing the seismic event comes from the equation of the frequency of the seismic event expressed in terms of the rupture velocity and the radius of the crack producing the seismic event, such that

$$2\pi * f = C/R, \quad 4.3$$

where f is the frequency, C is a function of the rupture velocity, and R is the radius of the crack (Aki and Richards, 2002). From equation 4.1, we find the frequency of a seismic event is inversely proportional to the size of the fracture (Aki and Richards, 2002).

We use fracture lengths on the order of 10^{-1} m for the laboratory (since our samples do not exceed 0.30 m in size), 10^2 m for hydraulic fracturing (Economides and Nolte, 2000; Jones and

Britt, 2009), and 10^3 m for magmatic fracturing (Burlini et al., 2007). For the dominant frequency, we use 10^4 Hz as these values divide high frequency and low frequency events in their respective environments (Table 1.1), 100 Hz for hydraulic fracturing (Eaton et al., 2013), and 5 Hz for magmatic fracturing (Zobin, 2012). We find that the field seismic events and the laboratory seismic events scale approximately by an order of magnitude of 3 for hydraulic fractures and 4 for volcanic events.

Table 4.1. The scaling between the laboratory (L), magmatic (V) and hydraulic fracturing (HF) environments based on the inverse relationship between the source dimension and the dominant frequency (Aki and Richards, 2002).

Dimension scaling	Values	Dominant Frequency scaling	Values	
			High frequency events	Low frequency events
$\frac{d_{HF}}{d_L}$	$\frac{10^2}{10^{-1}} = 10^3$	$\frac{f_L}{f_{HF}}$	$\frac{10^{5*}}{10^2} = 10^3$	$\frac{10^4}{10^1} = 10^3$
$\frac{d_v}{d_L}$	$\frac{10^3}{10^{-1}} = 10^4$	$\frac{f_L}{f_v}$	$\frac{10^{5*}}{10^1} = 10^4$	$\frac{10^4}{10^0} = 10^4$
$\frac{d_v}{d_{HF}}$	$\frac{10^3}{10^2} = 10^1$	$\frac{f_{HF}}{f_v}$	$\frac{10^2}{10^1} = 10^1$	$\frac{10^1}{10^0} = 10^1$

4.2 The microseismic event variations in experiment 1 and 2

Our results combined with the work of previous studies support the belief that hydraulic fracturing occurs predominantly aseismically. Aseismic deformation occurs without any detectable seismic events or with seismic events of very low amplitudes. For instance, the number of microseismic events occurring in experiment 2 is 7 times the number of events occurring in experiment 1 with the seismic events in experiment 1 with lower amplitudes than experiment 2 (Appendix E & D). Despite the differences in the number of microseismic events detected, in both experiments, we observe the initiation and propagation of the hydraulic

fracture. The detection of no seismic events or very few low amplitude events during hydraulic fracturing does not necessarily indicate that no deformation is taking place, it can mean that the energy released during deformation is not large enough to be detected clearly by the seismometers measuring the seismic response.

The results of the two experiments also show that most of the microseismic events generated in relation to the presence of pre-existing fractures or weaknesses within the medium that is being fractured. Recent hydraulic fracturing studies (Dahi Taleghani and Olson, 2014; Maxwell et al., 2015; Rutledge et al., 2015) attribute the generation of the microseismic events to a shear deformation mechanism caused by slip along natural fractures even before the hydraulic fracture intersects the natural fracture. However, in the field, the assumption is the microseismic events occur in direct relation to the propagating edge of the hydraulic fracture and thus the locations of the microseismic events are synonymous with the edge of the fracture (Maxwell et al., 2010).

The microseismic events in experiment 1 support the former argument and do not follow the growth of the hydraulic fracture with locations at or close to the fracture edge as expected during the propagation of the fracture. Instead, the events all occur close to the well bore where the likely cause of the microseismic events produced is the damage near the wellbore caused during the sample preparation (Section 1). The wellbore damage creates discontinuity surfaces (offsets) and shearing occurs as the offsets propagate along the fracture (Figure 3.20). The offsets become detached from the main surface of the fracture. The shearing mechanism on the discontinuity surfaces is the likely cause of the microseismic events especially since the locations of these events are found to occur near the start of the offset surfaces observed in the sample (Figures 3.8 & 3.20).

In the field in an area with pre-existing fractures and faults, the microseismic events are relatively higher in amplitude and tend to occur or move towards an area with pre-existing faults or that has a high density of natural fractures (Maxwell et al., 2011). For instance, in a case study on a three horizontal well hydraulic fracture treatment in the Montney formation in Canada, microseismic events with the highest amplitudes throughout the treatment, occur in the south east of the field close to a previously unknown fault (Maxwell et al., 2011). The microseismic event locations throughout the treatment in the case study tend to be more concentrated in the south east (Maxwell et al., 2011). In experiment 2, the microseismic events occur on the western side of the well (the side of the hydraulic fracture that grows toward the pre-existing fracture).

The events that occur in association with a pre-existing fracture are higher in amplitude because of the higher stresses involved in the interaction of the two fractures and also because the interaction can cause slip along the pre-existing fracture. In the case study, one of the conclusions is the microseismic events grow in the area towards the pre-existing fault because the presence of the pre-existing fault can lower the stresses in the parts of the medium closest to the pre-existing weakness causing more energy to be released as the fractures grow.

When there are no pre-existing fractures, the microseismic events occur with a lower amplitude and the microseismic events that occur mainly as a result of the energy released as the fracture grows. In experiment 2, there are no microseismic events detected on the eastern side of sample 2 where there is no direct interaction of the pre-existing fracture and the growing hydraulic fracture on this side of the sample. There may be microseismic events that occur but that cannot be detected above the noise level because of their low amplitude.

The events occurring have very low amplitudes because of the mechanism of the generation of the events which is mostly likely by aseismic tensile deformation associated with the growth of the hydraulic fracture.

4.3 The pressure variations in experiment 1 and 2

The pump pressure decline in experiment 2 does not show a smooth decline curve as expected and as observed in experiment 1. Although there is an overall decrease there are multiple fluctuations throughout pump pressure decline. The most likely reason for the oscillations in the pump pressure, is the alternating closing and opening of the growing fracture as it interacts with the pre-existing. The stresses acting at the tip of the growing fracture can be large enough to increase the shear stress acting on the pre-existing fracture (Dahi Taleghani and Olson, 2014). The increase in shear stress can debond the pre-existing fracture or cause slip along the fracture. The angle of approach of the hydraulic fracture, can affect whether complete debonding of the pre-existing fracture occurs, which leads to some areas along the fracture remaining cemented (Dahi Taleghani and Olson, 2014).

Areas on the pre-existing fracture that are still cemented can exert a higher stress on the growing fracture causing the fracture to close. As the created fracture closes, the width decreases causing the fluid pressure to increase. The increase in the fluid pressure occurs to overcome the closure stress in addition to the stresses induced by the interactions and decreases when the width increases allowing fluid flow through the fracture. In experiment 2, the growing fracture deflects into one side of pre-existing fracture because deflection into one side is the path of growth that releases the greatest amount of energy (Dahi-Taleghani and Olson, 2011; Dahi Taleghani and Olson, 2014).

Another possible explanation for the fluctuations in the decline of the pump pressure is because of the intersection of a pre-existing sealed fracture with a fluid-driven fracture (Sibson, 2000). The interaction of the two fractures causes the sealed fracture to debond and the fluid from the growing fracture enters the formerly sealed fracture. The reopened pre-existing fracture expands and contracts as the fluid fills the space created in the pre-existing fracture (Sibson, 2000). The expansion and contraction causes fluctuations in the fluid pressure which is known as fault valve action (Sibson, 2000). In our experiment, the fluctuations in pressure occurs before the growing fracture intersects or even grows close enough to the pre-existing fracture in our sample. Hence, fault valve action is not a viable explanation for the fluctuations.

4.4 Future Work

Repeatability is one of the major uncertainties for the generation of the seismic model using the laboratory experiments. The two experiments and in particular the experiment with the pre-existing fracture, should be repeated to see if the results are comparable. If they are the validity of the model will increase in the case of fracturing in an isotropic material.

Future experiments varying the dip angle of the pre-existing fracture, the distance of the pre-existing fracture from the borehole, and the number of pre-existing fractures are necessary in order to determine how the microseismicity and the characteristics of the microseismic events will change by these factors. The variations will help to increase the level of anisotropy in the laboratory experiments and thus will make the results more comparable to the anisotropic environment in the field.

Incorporation of the source mechanisms for all the events identified in the experiments should be included in analysis of the seismic data obtained from future experiments as the

source mechanisms will help to directly support or nullify the type of deformation that is expected in the model.

Further consideration should also be given to how the preparation of the sample might affect the stresses within the sample itself. For our experiments, we have assumed the samples to be completely isotropic after thermal annealing. It should be noted that the attachment of the well and sensors on the block may reintroduce stresses that may or may not affect fracture growth.

Chapter 5 Conclusions

Experiments using polymethyl methacrylate (PMMA) to simulate hydraulic fracturing can be used to provide good insight into the deformation and the microseismic events that occur during fracturing in the field and in magmatic settings. The microseismic events that occur during fluid-driven fracturing can be categorized into 6 groups based on the variations of their characteristics of frequency, amplitude and duration, and whether the events occur singly or in a cluster. High frequency events in the two experiments conducted have frequencies greater than 10,000 Hz and low frequency events have frequencies less than 10,000 Hz. The variations in the types of seismic events are dependent on where the source of the event originates and the mechanism producing the event. We can use previous studies on the growth of magmatic fractures and hydraulic fractures in the field, to have a better idea of how the different microseismic events may vary with the growth of the fractures and probable locations of the sources of the events.

The results from the two experiments support the idea that hydraulic fracturing is predominantly an aseismic process. Experiment 1, which involves sample 1 without a pre-existing fracture, has only 15 events with lower amplitudes than experiment 2, with the pre-existing fracture, which had 110 events. The microseismic events in experiment 1, also do not follow the growing fracture tip as expected but most likely occur in areas close to the wellbore that experienced damage during the preparation of the sample. The interaction of the growing fracture with the pre-existing fracture is the likely cause of the generation of most of the microseismic events in Experiment 2. The interaction of the two fractures results in higher amplitude events, events that occur in the direction of the pre-existing fractures or predominantly in areas where pre-existing fractures are present, and an overall higher

quantity of microseismic events occurring compared to the events that occur in media with little to no pre-existing fractures. Even before the two fractures intersect, microseismic events occur along the pre-existing fracture indicating that slip and debonding occurs because of the influence of the stresses associated with the growing fracture tip acting on the pre-existing fracture.

Pump pressure fluctuations occur in experiment 2 during the decline of the pump pressure also because of the interaction of the growing fracture with the pre-existing fracture. The interaction of the two fractures results in the increase and decrease of the width of the fracture. The opening and closing of the fracture periodically in turn affects the fluid pressure needed to continue the propagation of the fracture. The fluid pressure increases as the fracture closes, since the fracture requires a higher pressure to remain open and to propagate. When the fracture is open, more space is available for the fluid to flow into the fracture, and the fluid pressure falls.

Further work such as moment tensor inversion of the microseismic events, will assist in confirming the mechanisms producing the microseismic events and will help to solidify the change in the type of microseismic events with the temporal and spatial growth of the hydraulic fracture.

References

- Abdi, H. and L. J. Williams, 2010, Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics **2**(4): 433-459.
- Aimene, Y. E. and J. A. Nairn, 2014, Modeling Multiple Hydraulic Fractures Interacting with Natural Fractures Using the Material Point Method, Society of Petroleum Engineers.
- Aki, K. and P. G. Richards, 2002, Quantitative seismology.
- Baig, A. and T. Urbancic, 2010, Magnitude determination, event detectability, and assessing the effectiveness of microseismic monitoring programs in petroleum applications. CSEG Recorder **35**(2): 22-26.
- Baig, A. and T. Urbancic, 2010, Microseismic moment tensors: A path to understanding frac growth. The Leading Edge **29**(3): 320-324.
- Benhama, A., C. Cllet and M. Dubesset, 1988, STUDY AND APPLICATIONS OF SPATIAL DIRECTIONAL FILTERING IN THREE-COMPONENT RECORDINGS1. Geophysical Prospecting **36**(6): 591-613.
- Benson, P. M., S. Vinciguerra, P. G. Meredith and R. P. Young, 2008, Laboratory simulation of volcano seismicity. Science **322**(5899): 249-252.
- Boore, D. M. and J. Boatwright, 1984, Average body-wave radiation coefficients. Bulletin of the Seismological Society of America **74**(5): 1615-1621.
- Brenner, S. L. and A. Gudmundsson, 2004, Arrest and aperture variation of hydrofractures in layered reservoirs. Geological Society, London, Special Publications **231**(1): 117-128.
- Brune, J. N., 1970, Tectonic stress and the spectra of seismic shear waves from earthquakes. Journal of Geophysical Research **75**(26): 4997-5009.
- Bunger, A. P., 2008, A rigorous tool for evaluating the importance of viscous dissipation in sill formation: it's in the tip. Geological Society, London, Special Publications **304**(1): 71-81.
- Burlini, L., S. Vinciguerra, G. Di Toro, G. De Natale, P. Meredith and J.-P. Burg, 2007, Seismicity preceding volcanic eruptions: New experimental insights. Geology **35**(2): 183-186.
- Chaubey, Y. P., J. Li, A. Sen and P. K. Sen, 2012, A new smooth density estimator for non-negative random variables. Journal of the Indian Statistical Association **50**: 83-104.

- Chouet, B.,1986, Dynamics of a fluid-driven crack in three dimensions by the finite difference method. *Journal of Geophysical Research: Solid Earth* **91**(B14): 13967-13992.
- Cohen, J. K. and J. J. W. Stockwell,2013, CWP/SU: Seismic Un*x Release No. 43R4: an open source software package for seismic research and processing. Colorado School of Mines, Center for Wave Phenomena
- Dahi-Taleghani, A. and J. E. Olson,2011, Numerical modeling of multistranded-hydraulic-fracture propagation: Accounting for the interaction between induced and natural fractures. *SPE journal* **16**(03): 575-581.
- Dahi Taleghani, A.,2009, Analysis of hydraulic fracture propagation in fractured reservoirs: an improved model for the interaction between induced and natural fractures. Doctoral dissertation, University of Texas, Austin, <http://hdl.handle.net/2152/18381>.
- Dahi Taleghani, A. and J. E. Olson,2014, How natural fractures could affect hydraulic-fracture geometry. *SPE journal* **19**(01): 161-171.
- Das, I. and M. D. Zoback,2011, Long-period, long-duration seismic events during hydraulic fracture stimulation of a shale gas reservoir. *The Leading Edge* **30**(7): 778-786.
- Das, I. and M. D. Zoback,2013, Long-period long-duration seismic events during hydraulic stimulation of shale and tight-gas reservoirs — Part 2: Location and mechanisms. *Geophysics* **78**(6): KS109-KS117.
- Detournay, E.,2004, Propagation regimes of fluid-driven fractures in impermeable rocks. *International Journal of Geomechanics* **4**(1): 35-45.
- Detournay, E., A. Peirce and A. Bunger,2007. Viscosity-dominated hydraulic fractures. 1st Canada-US Rock Mechanics Symposium, American Rock Mechanics Association.
- Eaton, D., M. van der Baan, J.-B. Tary, B. Birkelo, N. Spriggs, S. Cutten and K. Pike,2013, Broadband microseismic observations from a Montney hydraulic fracture treatment, northeastern B.C., Canada. *Canadian Society of Exploration Geophysicists Recorder* **38**(03): 45-53.
- Economides, M. J. and K. G. Nolte,2000, *Reservoir Stimulation*, Wiley.
- Frash, L. P.,2007, Laboratory-scale study of hydraulic fracturing in heterogeneous media for enhanced geothermal systems and general well stimulation, Colorado School of Mines. Arthur Lakes Library.
- Galperin, Y. I.,1955, Azimuthal method of seismic acquisition. GosKomIzdat.

- Garagash, D. and E. Detournay, 2002. Viscosity-dominated regime of a fluid-driven fracture in an elastic medium. IUTAM Symposium on Analytical and Computational Fracture Mechanics of Non-Homogeneous Materials, Springer Netherlands.
- Graizer, V., 2009, The response to complex ground motions of seismometers with Galperin sensor configuration. Bulletin of the Seismological Society of America **99**(2B): 1366-1377.
- Grandi, S., R. V. Rao and M. N. Toksoz, 2002, Geomechanical modeling of in-situ stresses around a borehole, Massachusetts Institute of Technology. Earth Resources Laboratory.
- Han, L., J. Wong and J. Bancroft, 2010, Back propagation analysis for hypocenter location, CREWES Research Report, this volume.
- He, M.-Y. and J. W. Hutchinson, 1989, Crack deflection at an interface between dissimilar elastic materials. International Journal of Solids and Structures **25**(9): 1053-1067.
- IDEMAT, 2003. Materials Selection Guide Retrieved August 2014, 2014, from <http://www.matweb.com/search/DataSheet.aspx?MatGUID=2e5aa174e89d4f0293d29fc76582ac99&ckck=1>.
- Jones, J. R. and L. K. Britt, 2009, Design and Appraisal of Hydraulic Fractures, Society of Petroleum Engineers.
- Kim, G. H. and J. Y. Wang, 2011, Interpretation of Hydraulic Fracturing Pressure in Low-Permeability Gas Formations, Society of Petroleum Engineers.
- Klema, V. and A. Laub, 1980, The singular value decomposition: Its computation and some applications. IEEE Transactions on automatic control **25**(2): 164-176.
- Kraft, 2015. Ingredients for Kool-Aid Tropical Punch Drink Mix. Retrieved February 2015, from <http://www.kraftrecipes.com/products/kool-aid-tropical-punch-drink-m-930.aspx>.
- Lees, J. M., 2014, RSEIS: Seismic Time Series Analysis Tools.
- Liner, C. L., 2004, Elements of 3D Seismology, PennWell.
- Liu, A., 1996, Summary of stress-intensity factors. ASM International, Member/Customer Service Center, Materials Park, OH 44073-0002, USA, 1996.: 980-1000.
- Lorenzo, J. M., 2015, RHFM: R-based hydraulic fracturing microseismic tools.

M.I.T.,2014. Material Properties Database: PMMA. Retrieved August 2014, 2014, from <http://www.mit.edu/~6.777/matprops/pmma.htm>.

MAELABS,2011. Retrieved August 2014, 2014, from <http://maelabs.ucsd.edu/mae171/materialsdocs/LabWeek1,%20Modified,%20Jan%202011-1.pdf>.

Maercklin, N.,2001, SUPOLAR and SUPOFIL: SU programs for polarization analysis and filtering of three-component data.

Maimon, O., V. Lyakhovsky, O. Melnik and O. Navon,2012, The propagation of a dyke driven by gas-saturated magma. *Geophysical Journal International* **189**(2): 956-966.

Maxwell, S., J. Rutledge, R. Jones and M. Fehler,2010, Petroleum reservoir characterization using downhole microseismic monitoring. *Geophysics* **75**(5): 75A129-175A137.

Maxwell, S. C.,2011, What does Microseismic Tell Us About Hydraulic Fracture Deformation. *Canadian Society of Exploration Geophysicists Recorder* **36**(08): 30-45.

Maxwell, S. C.,2014, Microseismic Imaging of Hydraulic Fracturing: Improved Engineering of Unconventional Shale Reservoirs. *Society of Exploration Geophysicists 2014 Distinguished Instructor short Course, Distinguished Instructor Series*(17).

Maxwell, S. C., D. Cho, T. L. Pope, M. Jones, C. L. Cipolla, M. G. Mack, F. Henery, M. Norton and J. A. Leonard,2011, Enhanced Reservoir Characterization Using Hydraulic Fracture Microseismicity, *Society of Petroleum Engineers*.

Maxwell, S. C., D. Chorney and S. D. Goodfellow,2015, Microseismic geomechanics of hydraulic-fracture networks: Insights into mechanisms of microseismic sources. *The Leading Edge* **34**(8): 904-910.

Neuberg, J. W., H. Tuffen, L. Collier, D. Green, T. Powell and D. Dingwell,2006, The trigger mechanism of low-frequency earthquakes on Montserrat. *Journal of Volcanology and Geothermal Research* **153**(1): 37-50.

Nolte, K. G. and M. B. Smith,1981, Interpretation of Fracturing Pressures.

R Core Team,2013, R: A language and environment for statistical computing. . Vienna, Austria, R Foundation for Statistical Computing.

Rasband, W. S.,1997, ImageJ. Bethesda, Maryland, USA, US. National Institutes of Health.

- Richardson, M.,2009, Principal Component Analysis. Johns Hopkins University.
- RStudio Team,2013, Rstudio: Integrated Development for R. Boston, MA, RStudio Inc.
- Rutledge, J., X. Yu and S. Leaney,2015, Microseismic shearing driven by hydraulic-fracture opening: An interpretation of source-mechanism trends. *The Leading Edge* **34**(8): 926-934.
- Saenger, E. H., S. M. Schmalholz, M.-A. Lambert, T. T. Nguyen, A. Torres, S. Metzger, R. M. Habiger, T. Müller, S. Rentsch and E. Méndez-Hernández,2009, A passive seismic survey over a gas field: Analysis of low-frequency anomalies. *Geophysics* **74**(2): O29-O40.
- Salem,2014. Retrieved August 2014, 2014, from <http://www.salemball.com/acrylic.htm>.
- Scott, D. W.,2015, Kernel Density Estimators. *Multivariate Density Estimation*, John Wiley & Sons, Inc: 137-216.
- Shelly, D. R., G. C. Beroza and S. Ide,2007, Non-volcanic tremor and low-frequency earthquake swarms. *Nature* **446**(7133): 305-307.
- Sibson, R.,1977, Fault rocks and fault mechanisms. *Journal of the Geological Society* **133**(3): 191-213.
- Sibson, R. H.,1996, Structural permeability of fluid-driven fault-fracture meshes. *Journal of Structural Geology* **18**(8): 1031-1042.
- Sibson, R. H.,2000, Fluid involvement in normal faulting. *Journal of Geodynamics* **29**(3): 469-499.
- Stork, A. L., J. P. Verdon and J. M. Kendall,2014, The robustness of seismic moment and magnitudes estimated using spectral analysis. *Geophysical Prospecting* **62**(4): 862-878.
- Tary, J. B., M. van der Baan, B. Sutherland and D. W. Eaton,2014, Characteristics of fluid-induced resonances observed during microseismic monitoring. *Journal of Geophysical Research: Solid Earth* **119**(11): 8207-8222.
- Townsend, B.,2014, Symmetric Triaxial Seismometers.
- van der Baan, M., D. Eaton and M. Dusseault,2013, Microseismic Monitoring Developments in Hydraulic Fracture Stimulation.
- van der Pluijm, B. A. and S. Marshak,2004, *Earth Structure*

An Introduction to Structural Geology and Tectonics, W. W. Norton and Company.

Warpinski, N.,2009, Microseismic Monitoring: Inside and Out.

Warpinski, N. R.,2013. Understanding hydraulic fracture growth, effectiveness, and safety through microseismic monitoring. ISRM International Conference for Effective and Sustainable Hydraulic Fracturing, International Society for Rock Mechanics.

Warpinski, N. R., P. T. Branagan, R. E. Peterson, S. L. Wolhart and J. E. Uhl,1998, Mapping Hydraulic Fracture Growth and Geometry Using Microseismic Events Detected by a Wireline Retrievable Accelerometer Array, Society of Petroleum Engineers.

Warpinski, N. R. and L. W. Teufel,1987, Influence of Geologic Discontinuities on Hydraulic Fracture Propagation (includes associated papers 17011 and 17074).

Wassermann, J.,2002, Volcano seismology. IASPEI new manual of seismological observatory practice **1**(13): 662-703.

Wu, R., L. N. Germanovich, P. E. Van Dyke and R. P. Lowell,2007, Thermal technique for controlling hydraulic fractures. Journal of Geophysical Research: Solid Earth **112**(B5).

Zobin, V. M.,2012, 2 - Seismicity at Volcanoes. Introduction to Volcanic Seismology (Second Edition). V. M. Zobin. Oxford, Elsevier: 9-28.

Zobin, V. M.,2012, 3 - Fundamentals of Volcanic Seismology. Introduction to Volcanic Seismology (Second Edition). V. M. Zobin. Oxford, Elsevier: 29-48.

Zobin, V. M.,2012, 4 - Origin of Volcano-Tectonic Earthquakes. Introduction to Volcanic Seismology (Second Edition). V. M. Zobin. Oxford, Elsevier: 49-65.

Zobin, V. M.,2012, 12 - Volcanic Tremor. Introduction to Volcanic Seismology (Second Edition). V. M. Zobin. Oxford, Elsevier: 237-259.

Zuppann, C. W. and J. C. Steinmetz,2014, Hydraulic Fracturing: An Indiana assessment, Indiana Geological Survey.

Appendix

A. Non-dimensional toughness parameters

Table A.1 Fracture toughness, K , the Young's modulus, E , and the Poisson's ratio, ν , for polycarbonate and the upper and lower boundaries of shale, and the fluid viscosity, μ , and the flow rate, Q_0 of the fracturing fluid used to fracture these three different materials. The values listed in this table are from the personal communication with Dr. Juan Lorenzo and *Dr. Arash Dahi-Taleghani (Table 2.1 contains properties for PMMA)

Material	$K_{IC} (Pa\sqrt{m})$	$E (Pa)$	ν	$\mu (cP)$	$Q_0 (liters/min)$
Polycarbonate (lexan)	1.05×10^6	2.3×10^9	0.35	92280	4×10^{-6}
Shale (lower)*	1×10^6	4×10^9	0.2	1	2.12×10^{-3}
Shale (upper)*	1.1×10^7	1.8×10^{10}	0.35	4	2.12×10^{-3}

Table A.2. The parameters used in the calculations of non-dimensional toughness in order to determine the more suitable material to use as the samples in the experiments between PMMA and polycarbonate. The values listed in this table are from the personal communication with Dr. Juan Lorenzo and *Dr. Arash Dahi-Taleghani.

Material	K'	E'	μ'	Q'_0^*	\tilde{K}
Polymethyl methacrylate	3351115.155	3.56×10^9	1107360	1.33×10^{-9}	1.174
Polycarbonate (lexan)	5263919.377	2.62×10^9	1107360	8.33×10^{-9}	2.60718
Shale (lower)	3191538.243	4.17×10^9	12	0.00212	0.48726
Shale (upper)	35106920.68	2.05×10^{10}	48	0.00212	1.14678

B. Sample preparation procedure

1. Materials and equipment needed for sample preparation

- | | |
|---|---|
| 1. Polymethyl methacrylate (PMMA)
rectangular blocks | 8. Polarization chamber (wooden box,
black cloth, 2 polarization sheets) |
| 2. Drill | 9. Sensor mount (PMMA cylinder) |
| 3. Sandpaper | 10. 24 Piezoelectric sensors |
| 4. Glue | 11. 24 cables |
| 5. Oven | 12. 3 X 12-channel amplifier |
| 6. Valve | 13. Well pressure sensor |
| 7. Vices | |

2. Procedure

1. We use 2 PMMA blocks as our samples in the two experiments. We carry the blocks to the LSU Machine shop to be machined so that our samples have parallel sides to an accuracy of ± 0.001 inches (Section 2.2 & Table 2.2 for sample descriptions).
2. We polish and sand the samples after the machining, because the samples are left opaque and dull.
3. We create Sample 2, used in experiment 2, from two PMMA blocks each one having on edge at an angle of $60^\circ \pm 0.083^\circ$ from the horizontal. We roughen a surface along the angled sides of the two blocks by using sandpaper and a scarping with a knife because the roughened surface provides more traction which helps to increase the hold of the glue. To bind the blocks we use J-B Weld as a glue (specifications in Table

2.3). To hold the sample in place while the glue sets, we use vices to hold the two blocks together and to keep them in place.



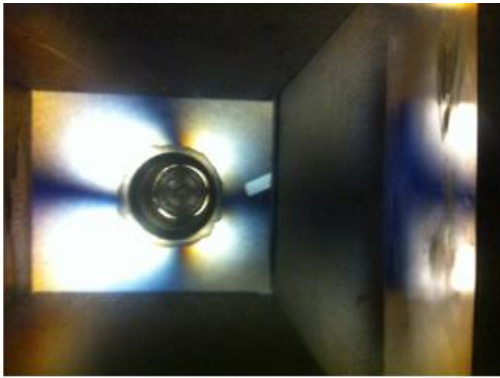
Figure B.1. The PMMA blocks during the sample preparation. Two blocks on the left of the figure already contain the wells and pre-cracks. Two blocks on the right of the figure show one side each cut at angle of 60° from the horizontal before the two blocks are bound to create sample 2.



Figure B.2. Preparation of sample 2. a) The finished product, Sample two, the glue completely dries and sets. b) JB-Weld mixture that is applied to the separate blocks in order to bind the blocks. After the glue is applied, vices hold the sample together to make sure that the two blocks stay in place while the glue dries.

4. We drill a borehole in the center of each sample with a diameter 0.0191 m to a depth of 0.05715 m and we create a pre-crack at a depth of approximately half of the sample's height (0.038 m) with a diameter no greater than ± 0.02 m from the center of the borehole.
5. In order to reduce the internal stresses caused by the cutting and drilling of the samples, we conduct thermal treatment on each sample. We place each sample in an oven for 36 hours at a temperature of 80 °C. The blocks cool at a rate of ~ 15 °C per hour for 24 hours after they are treated in the oven. To verify that the samples are annealed sufficiently, we examine the blocks before and after under cross-polarized light within a dark chamber. The dark chamber consists of a black wooden box covered with a black cloth to block unwanted light and reflections from entering. We place the sample in the dark chamber and we use a fluorescent light below the sample for illumination and two polarization film sheets (one above the sample and one below the sample) so that the sample can be viewed under cross polarized light. The lower sheet prevents light from vibrating in one direction (N-S or E-W) and the upper sheet prevents light from vibrating in the other direction (Imperial College London, 2013). If the sample is anisotropic, that is if the stresses within the samples are not equal, then the incident light rays will be split into two and when recombined at the top polarization film can produce interference patterns (Imperial College London, 2013). Each sample shows a birefringence cross after thermal annealing and before the addition of the valve to the well (**Error! Reference source not found.**).

a)



b)

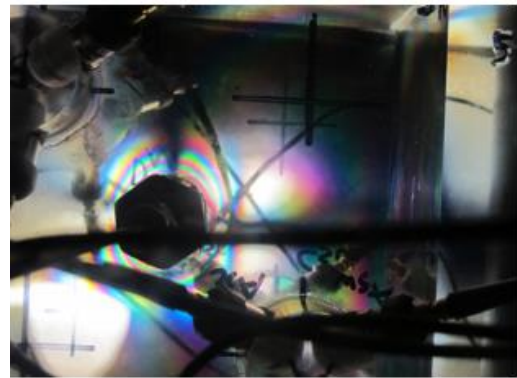


Figure B.3. The interference figures of sample 1 after the drilling of the well and thermal annealing (a) and after the valve has been added to the wellbore (b). These pictures show that after rigorous thermal annealing, the induced stresses are not fully removed from the samples.

6. To prevent air bubbles from being trapped within the borehole and sample during the experiment, we fill the borehole with fracturing fluid before the start of the experiment. To secure the fluid, temporarily, in the borehole we place a valve at the top of the borehole. The valve also serves as a connection from the tube of the pump to the sample.

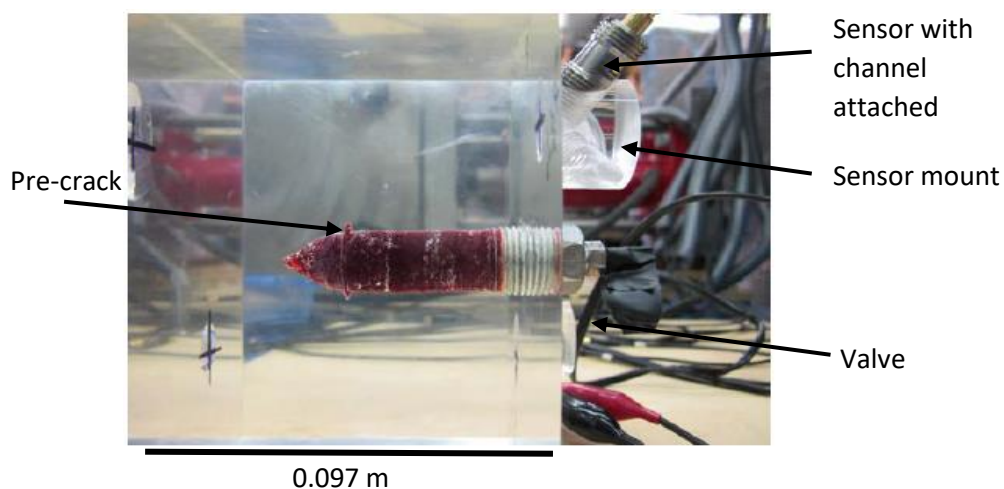


Figure B.4. Sample 1 oriented with its top to the right of the picture. 0.0191 m diameter borehole is filled with fracturing fluid (pink) and capped with valve to which the pump is attached for the experiment. The pre-crack is visible approximately at half of the sample's height (~ 0.038 m).

7. We add sensor mounts to the top and bottom of the sample (using glue). We use small cylindrical PMMA mounts, 0.0381 m in diameter and 0.0254 m in height as the sensor mount. Each station mount is made with three drilled holes in a Galperin arrangement, that is, three components are orthogonal to each other (Grazier, 2009; Galperin, 1955). We fit three sensors into each sensor mount. We then connect the sensors to the channel amplifier and power supply (Figure B.5).

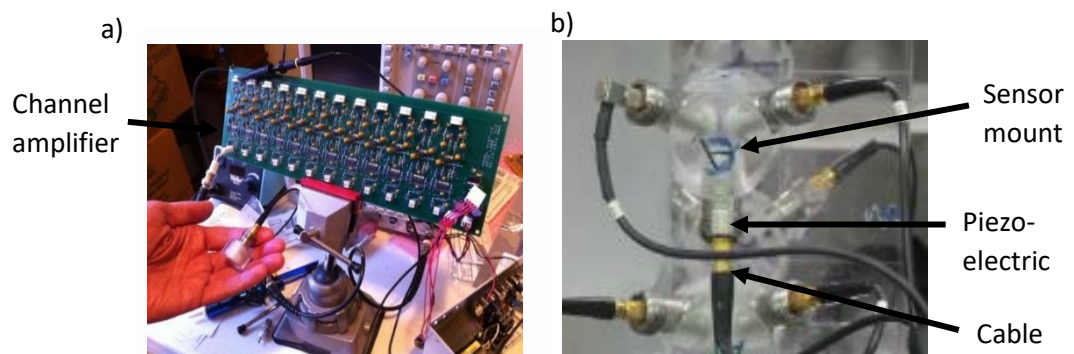


Figure B.5. (a) A 12-Channel amplifier and power supply (b) Set up of each station on the samples. We attach four sensor mounts to the top of the sample and four to the bottom of the sample. Each sensor mount has three holders for the sensors. We attach the sensors to each station and a cable, which acts a channel connecting each sensor to the acquisition system, to each sensor.

8. We test each of the sensors before the experiment by connecting each sensor via its channel to a voltmeter. We gently tap the samples and monitor the response on the voltmeter in order to verify that the sensors are in fact working.
9. Before the sample is placed in the biaxial press, the P-Setra Pressure Transducer is attached to the top of the well.



Figure B.6. Inline pressure sensor (Industrial Pressure Transducer Model 522 from Setra) that is attached to the top of the well.

C. Fracturing Fluid Preparation

1. Materials and Equipment needed:

- | | |
|------------------------------|-------------------------------|
| a) Sugar (sucrose) | g) Handheld mixer |
| b) Water | h) Viscometer |
| c) Glucose | i) Triple beam balance/mass |
| d) Kool-Aid (dye) | balance |
| e) Container for mixing | j) Heating device (microwave) |
| f) Beaker/measuring cylinder | |

2. Procedure

2.1 We measure 560 g of sugar (sucrose) using the beam balance and we add 560 ml of water to it in the mixing container.

2.2 We add approximately 125 g ($\frac{1}{4}$ of the total amount of the initial amount of the sugar) to the water.

2.3 We heat the mixture for 30 seconds in the microwave before mixing thoroughly with a handheld mixer.

2.4 After the solid dissolves completely, we proceed to measure the viscosity using the Brookfield Viscometer.

2.5 The size of the spindle we use for the viscometer is "S61" and we place this into the container with the fracturing fluid.

2.6 We adjust the spin rate until we obtain an accuracy as close as possible to 100 % (the cut off percentage we set for our measurements is 90 % and above). We record the viscosity in cP (centipoise).

2.7 We continue to add small amounts of the sugar until the entire 560 g of sugar are added to the water and repeat steps 2.3 to 2.6.

2.8 After 560 g is added, we add glucose (in increments of 1000 -2000 g) and the sugar (in increments of 25 g) and repeat steps 2.3 to 2.6. We continue adding the glucose and sugar in this way until we reach of ratio of 80:20 (sugar to glucose). From previous experiments and tests in the mixing of the fracturing fluid, this ratio helps to prevent the crystallization of the sugar and glucose out of solution and allows for easier mixing. We continue adding glucose and sugar until our desired viscosity of 92280 cp (the viscosity used for both experiments) is reached.

D. Experimental Procedure

1. Materials

(Table 2.3 & Table 2.4)

2. Procedure

1. The sample is placed in the middle of the biaxial press (Figure 2.4 & Figure D.1). The dimensions of the sample, does not allow for the biaxial press to close exactly on the sample so we use 4 smaller PMMA blocks between the sample and the biaxial press. The 4 block ensure that the sample remains in place and the stress is evenly transmitted from the plates to the samples. We adjust the pressure applied by the press and measure how much is applied by using the pressure gauges. We want to ensure that 1000 psi of pressure is applied in the two horizontal directions.



Figure D.1. Biaxial press where the sample is in the middle of the four steel plates. $\frac{1}{8}$ tubing connects the pump to the well to direct the flow of the fracturing fluid from the pump to the well.

2. We install the cables connecting the sensors on the samples to the seismic acquisition cards, where the data is recorded, on the sensors on the block.

3. We place the camera above the sample and we position the LED light so that when it is lit, it is in the field of view of the camera.
4. A fluorescent light, below the sample and a black cloth covering the sample and the biaxial press provide a darkened environment so that dye within the fracturing fluid illuminates during the experiment.
5. We test the acquisition equipment before the start of the experiment to ensure that the sensors and pressure meters are working and the readings are reasonable.
6. We start recording on the camera, the well-head pressure meter, the seismic data and the pump. We begin pumping at a rate of 4 $\mu\text{l}/\text{min}$ which we maintain throughout the course of experiment 1. For experiment 2, we start off at a rate of 4 $\mu\text{l}/\text{min}$, and then at 10,658 seconds we increase the rate to 8 $\mu\text{l}/\text{min}$ and at 16,358 seconds we increase the rate to 12 $\mu\text{l}/\text{min}$.
7. At intervals of time between 5 and 15 minutes, we produce synchronization signals. A single frequency square wave creates a signal in the three acquisition systems such that a red LED light turns on in the field of view of the camera, capacitive coupling creates a voltage in the pump pressure recording, and inductive coupling creates a voltage in the well-pressure and microseismic data recording.

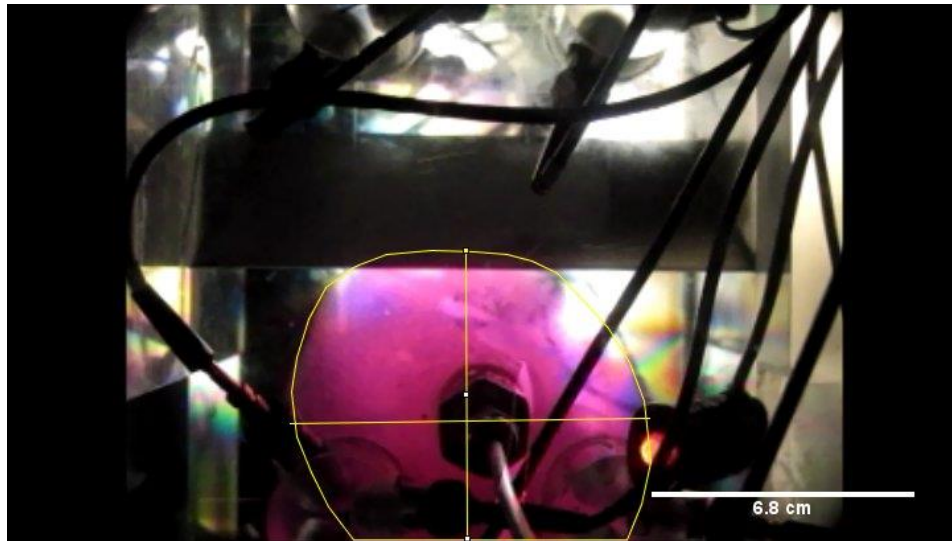


Figure D.2. Still image of the fracture during experiment 2 showing the outline of the fracture (yellow line) and the red LED light on during a synchronization signal.

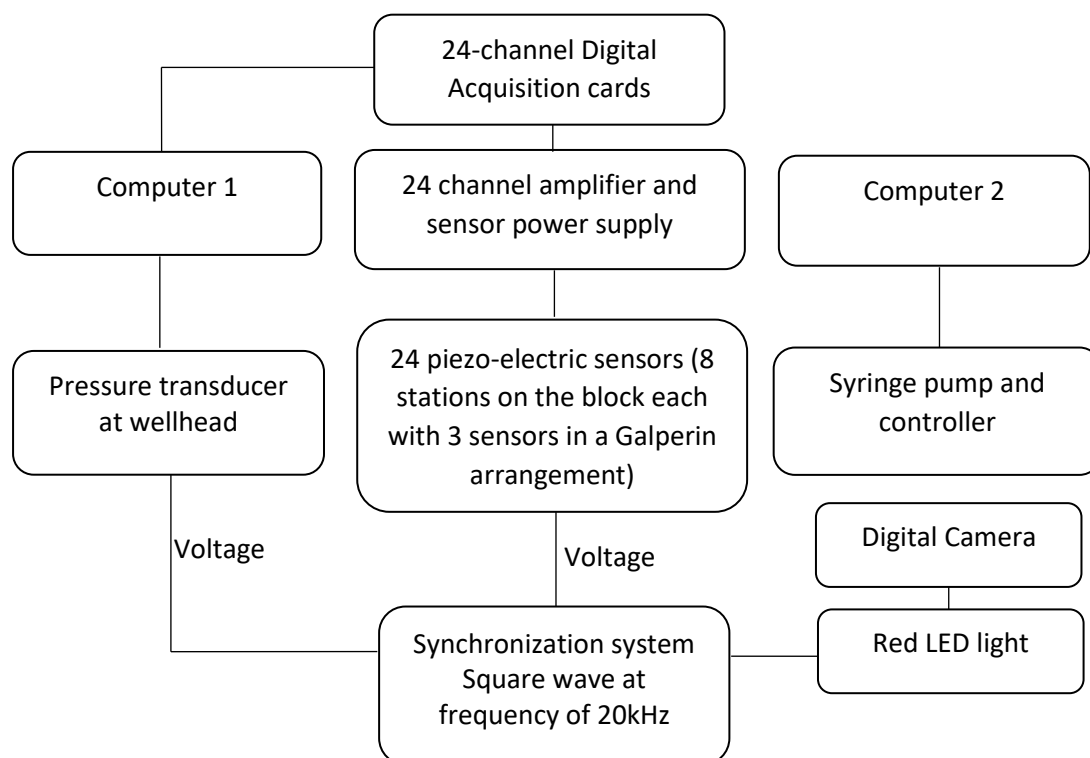


Figure D.3. Schematics of the acquisition system for the experiments

E. Characteristics and locations for the microseismic events in Experiment 1

Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	Location		
				x coordinate (m)	y coordinate (m)	z coordinate (m)
1	7657.89989	17603.86667	99	0.0794 ± 0.006	0.0687 ± 0.1453	-0.0719 ± 0.0047
2	7672.978122	16695.33333	428	0.07 ± 0.0039	0.0832 ± 0.1697	-0.0828 ± 0.0038
3	7683.60493	17537.55556	216	0.0816 ± 0.0036	0.0718 ± 0.1076	-0.0402 ± 0.0044
4	7690.77932	3655.555556	1130	0.0781 ± 0.0049	0.0793 ± 0.1283	-0.049 ± 0.0115
5	7699.3911	24844	112	0.0828 ± 0.0051	0.071 ± 0.1291	-0.0579 ± 0.0081
6	7757.8951	15409.33333	1054	0.0772 ± 0.0053	0.0693 ± 0.1165	-0.0475 ± 0.0061
7	7802.84	17895.65927	223	0.0753 ± 0.0046	0.0752 ± 0.1538	-0.0785 ± 0.0082
8	7806.687832	17011.66667	159	0.0821 ± 0.0047	0.0761 ± 0.1491	-0.077 ± 0.004
9	7945.3488	37739.66667	872	0.0702 ± 0.0087	0.0776 ± 0.1379	-0.0558 ± 0.0046
10	7947.73393	15315.68873	1449	0.0689 ± 0.0053	0.0865 ± 0.1511	-0.0648 ± 0.0057
11	8899.739417	17838.28571	136	0.0715 ± 0.0065	0.0804 ± 0.1307	-0.0551 ± 0.0049
12	9818.57571	17636.83333	187	0.0705 ± 0.0063	0.0739 ± 0.1223	-0.0481 ± 0.0071
13	10830.82209	17901.33333	266	0.077 ± 0.0063	0.0759 ± 0.1365	-0.059 ± 0.0069
14	10863.8854	17872.53439	94	0.0761 ± 0.0058	0.0808 ± 0.1349	-0.0596 ± 0.0055
15	11108.41203	17629.66667	164	NA	NA	NA

F. Characteristics and locations for the the microsesimic events in Experiment 2						
Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	x coordinate (m)	Locations y coordinate (m)	z coordinate (m)
1	285.8036	15310.62901	769	0.1423 ± 0.0107	0.0719 ± 0.1228	-0.0461 ± 0.0048
2	293.5068	3942.666667	32552	$0.1325 \pm 9e-04$	0.0619 ± 0.1128	-0.0405 ± 0.0104
3	302.1598	15877.71046	1842	0.1277 ± -0.0039	0.0644 ± 0.1153	-0.0581 ± -0.0072
4	306.7501	16026.83333	434	0.1508 ± 0.0192	0.0602 ± 0.1111	-0.0457 ± 0.0052
5	314.1624	14802.60112	426	0.1422 ± 0.0106	0.079 ± 0.1298	-0.0459 ± 0.005
6	325.3152	16044.59268	1215	0.1374 ± 0.0058	0.0796 ± 0.1305	-0.0448 ± 0.0061
7	330.6325	15823.44416	3059	0.1337 ± 0.0021	0.0724 ± 0.1233	$-0.0503 \pm 6e-04$
8	348.91344	4477	16181	0.1344 ± 0.0028	0.0608 ± 0.1117	-0.0393 ± 0.0116
9	2499.308	14188.62963	255	0.1199 ± -0.0117	0.0592 ± 0.1101	-0.0319 ± 0.019
10	5441.4025	4469	420	0.1341 ± 0.0025	0.0741 ± 0.1249	-0.065 ± -0.0141
11	6020.6444	22049.27536	271	0.1294 ± -0.0022	0.0561 ± 0.107	-0.0525 ± -0.0016
12	6058.84915	14752.55689	215	0.1121 ± -0.0195	0.0728 ± 0.1237	-0.0468 ± 0.0041
13	6063.2376	36338.4	135	0.1241 ± -0.0075	0.0696 ± 0.1205	-0.0488 ± 0.0021
14	6078.36624	14789.33333	137	0.11 ± -0.0216	0.0797 ± 0.1306	-0.0414 ± 0.0095
15	6118.128345	14141.6835	371	0.1193 ± -0.0123	0.073 ± 0.1239	-0.047 ± 0.0039
16	6190.06672	13147.60043	374	0.1174 ± -0.0141	0.0655 ± 0.1163	-0.0458 ± 0.0051
17	6193.3096	13242.26712	206	0.1265 ± -0.0051	0.0743 ± 0.1252	-0.0404 ± 0.0105
18	6265.0558	14490.23569	300	0.1116 ± -0.02	0.0755 ± 0.1264	-0.0497 ± 0.0012
19	6957.11596	15275.91667	270	0.1252 ± -0.0064	0.0615 ± 0.1124	-0.0523 ± -0.0014
20	7021.96879	4325.444444	282	0.1145 ± -0.0171	0.0708 ± 0.1217	-0.0435 ± 0.0074
21	7421.60717	4346.5	258	0.1231 ± -0.0085	0.0665 ± 0.1174	$-0.0513 \pm -5e-04$
22	7674.93919	4323.83133	569	0.1211 ± -0.0105	0.0717 ± 0.1226	-0.0394 ± 0.0114
23	7708.09225	15478.77066	2667	0.1056 ± -0.026	0.0714 ± 0.1223	-0.0414 ± 0.0095
24	8406.32095	4566.733333	212	0.1225 ± -0.0091	0.067 ± 0.1179	-0.0466 ± 0.0043

25	8437.22768	15686.46667	253	0.1238 ± -0.0077	0.0602 ± 0.1111	-0.0557 ± -0.0049
Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	x coordinate (m)	Locations y coordinate (m)	z coordinate (m)
26	8612.86416	4472.266667	171	0.1071 ± -0.0245	0.0689 ± 0.1197	-0.0434 ± 0.0074
27	9207.4428	4331.102662	440	0.1173 ± -0.0143	0.0661 ± 0.1169	-0.0588 ± -0.008
28	9215.8697	4105.866667	773	0.1144 ± -0.0172	0.0691 ± 0.1199	-0.0562 ± -0.0053
29	9228.5779	4027.13615	2942	0.1243 ± -0.0073	0.0683 ± 0.1192	-0.0524 ± -0.0015
30	9239.197081	3978.529063	6976	0.1137 ± -0.0178	0.0648 ± 0.1157	-0.0449 ± 0.006
31	9239.5659	4107	1827	0.1118 ± -0.0197	0.059 ± 0.1099	-0.0223 ± 0.0285
32	9436.08048	14823.84743	280	0.1162 ± -0.0154	0.0689 ± 0.1197	-0.0516 ± -7e-04
33	9495.99504	15759.26667	413	0.1127 ± -0.0189	0.0681 ± 0.1189	-0.044 ± 0.0069
34	10449.90608	12717.63158	191	0.119 ± -0.0126	0.0589 ± 0.1097	-0.0453 ± 0.0055
35	10516.2977	14425.8114	1183	0.1254 ± -0.0061	0.0632 ± 0.1141	-0.0497 ± 0.0012
36	10577.1209	14708.49673	310	0.1145 ± -0.0171	0.059 ± 0.1099	-0.04 ± 0.0109
37	10964.6264	11847.07672	169	0.117 ± -0.0146	0.0729 ± 0.1237	-0.0495 ± 0.0014
38	10966.32976	13829.80263	153	0.0979 ± -0.0337	0.0616 ± 0.1125	-0.0493 ± 0.0016
39	10974.19408	15754.22222	317	0.1146 ± -0.0169	0.0697 ± 0.1206	-0.0458 ± 0.0051
40	11028.9643	4364.380952	4139	0.1194 ± -0.0121	0.0704 ± 0.1213	-0.0531 ± -0.0022
41	11046.08516	15600.83333	284	0.117 ± -0.0146	0.0737 ± 0.1246	-0.0432 ± 0.0077
42	11053.4069	4763.061224	235	0.1101 ± -0.0215	0.055 ± 0.1059	-0.0447 ± 0.0062
43	11088.6202	3092.666667	353	0.1033 ± -0.0283	0.0636 ± 0.1145	-0.0383 ± 0.0126
44	11098.8545	3928.533333	735	0.1141 ± -0.0175	0.0568 ± 0.1076	-0.0355 ± 0.0154
45	11111.6309	14954.41169	946	0.1078 ± -0.0238	0.0639 ± 0.1148	-0.0406 ± 0.0102
46	11127.35468	20907.58772	513	0.1224 ± -0.0092	0.0748 ± 0.1257	-0.0229 ± 0.028
47	11176.1972	4344.866667	402	0.1152 ± -0.0164	0.079 ± 0.1298	-0.0464 ± 0.0045
48	11204.9072	3888.610039	301	0.1137 ± -0.0179	0.075 ± 0.1259	-0.0424 ± 0.0085
49	11332.6083	15185.38961	166	0.1113 ± -0.0202	0.0783 ± 0.1292	-0.0384 ± 0.0124
50	11442.90304	4625.666667	267	0.1151 ± -0.0165	0.065 ± 0.1159	-0.054 ± -0.0031

Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	Locations		
				x coordinate (m)	y coordinate (m)	z coordinate (m)
51	11474.36032	15774.66667	128	0.1222 ± -0.0094	0.0664 ± 0.1173	-0.0637 ± -0.0129
52	11587.08224	12878.67535	176	0.1248 ± -0.0068	0.0762 ± 0.1271	-0.0377 ± 0.0132
53	12319.29272	15563.69164	273	0.1235 ± -0.0081	0.0605 ± 0.1114	-0.044 ± 0.0069
54	12341.1233	4051.244444	1429	0.1113 ± -0.0203	0.0726 ± 0.1234	-0.0417 ± 0.0092
55	12353.5645	14610.4918	262	0.1078 ± -0.0238	0.0735 ± 0.1243	-0.041 ± 0.0099
56	12377.9632	4047.777778	677	0.1064 ± -0.0251	0.0721 ± 0.1229	-0.0412 ± 0.0097
57	12384	13369.07709	183	0.1074 ± -0.0242	0.0715 ± 0.1223	-0.0608 ± -0.0099
58	12569.9147	4502.666667	449	0.1269 ± -0.0047	0.0631 ± 0.114	-0.0461 ± 0.0048
59	12678.55952	15367.39394	169	0.1265 ± -0.0051	0.0772 ± 0.1281	-0.0548 ± -0.004
60	12808.1638	4369.333333	290	0.1251 ± -0.0064	0.0582 ± 0.1091	-0.0374 ± 0.0135
61	12827.4704	14897.19745	466	0.1094 ± -0.0221	0.0654 ± 0.1163	-0.0519 ± -0.001
62	12841.6514	15727.3372	314	0.1051 ± -0.0265	0.0726 ± 0.1234	-0.0694 ± -0.0186
63	12943.3858	4512.820513	330	0.1221 ± -0.0095	0.0946 ± 0.1455	-0.0504 ± 5e-04
64	13005.8782	15742.61328	2852	0.1132 ± -0.0184	0.0736 ± 0.1244	-0.0492 ± 0.0017
65	13046.5328	15081.55431	643	0.115 ± -0.0166	0.0647 ± 0.1156	-0.045 ± 0.0059
66	13068.0177	4330.666667	7758	0.1075 ± -0.0241	0.0548 ± 0.1057	-0.0394 ± 0.0115
67	13249.2824	21839.46667	286	0.1072 ± -0.0244	0.0668 ± 0.1177	-0.0545 ± -0.0036
68	13394.7314	3403.505933	205	0.1121 ± -0.0195	0.0693 ± 0.1201	-0.0497 ± 0.0012
69	13739.6856	14820.03922	185	0.1158 ± -0.0157	0.0865 ± 0.1374	-0.048 ± 0.0029
70	13845.3044	14037.81782	202	0.1062 ± -0.0254	0.0833 ± 0.1342	-0.0544 ± -0.0035
71	14160.9012	13671.01101	160	0.1226 ± -0.009	0.0736 ± 0.1244	-0.0488 ± 0.0021
72	14184.1001	4344.046057	1694	0.1059 ± -0.0257	0.0702 ± 0.1211	-0.0604 ± -0.0095
73	14718.70368	15388.63314	201	0.1047 ± -0.0269	0.0592 ± 0.1101	-0.0457 ± 0.0051
74	14798.8269	14938.06667	691	0.1234 ± -0.0082	0.0733 ± 0.1242	-0.0301 ± 0.0208
75	14944.7882	16202.19409	780	0.1132 ± -0.0184	0.0709 ± 0.1218	-0.0403 ± 0.0106

Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	Locations		
				x coordinate (m)	y coordinate (m)	z coordinate (m)
76	14949.8466	15329.2827	401	0.1189 ± -0.0127	0.0703 ± 0.1212	-0.0299 ± 0.0209
77	15028.5465	15089.59212	155	0.1075 ± -0.0241	0.0648 ± 0.1156	-0.0462 ± 0.0047
78	15032.6501	15391.13924	527	0.133 ± 0.0015	0.0742 ± 0.125	-0.0367 ± 0.0142
79	15039.1438	14268.52321	145	0.1212 ± -0.0104	0.0704 ± 0.1213	-0.0546 ± -0.0037
80	15043.76224	15563.45992	227	0.12 ± -0.0116	0.0709 ± 0.1218	-0.047 ± 0.0039
81	15054.5245	15764.69761	232	0.1014 ± -0.0302	0.0808 ± 0.1317	-0.0549 ± -0.004
82	15062.396	15104.92264	206	0.1266 ± -0.005	0.0652 ± 0.116	-0.0557 ± -0.0048
83	15073.0599	15352.29255	250	0.1074 ± -0.0242	0.0704 ± 0.1213	-0.0405 ± 0.0103
84	15081.3267	14894.0647	270	0.1095 ± -0.022	0.0637 ± 0.1146	-0.0594 ± -0.0086
85	15090.4173	15581.26582	593	0.116 ± -0.0155	0.0709 ± 0.1218	-0.0462 ± 0.0047
86	15091.263	4028.571429	2943	0.1159 ± -0.0157	0.0647 ± 0.1155	-0.045 ± 0.0059
87	15093.6986	4221.209564	2849	0.1029 ± -0.0287	0.0831 ± 0.1339	-0.0445 ± 0.0064
88	15097.0644	4465.988701	3331	0.1133 ± -0.0183	0.0631 ± 0.114	-0.0394 ± 0.0115
89	15106.9184	4982.4	307	0.1317 ± 1e-04	0.0681 ± 0.119	-0.0444 ± 0.0065
90	15142.9029	13408.34339	273	0.1062 ± -0.0254	0.0667 ± 0.1176	-0.0426 ± 0.0082
91	15164.3758	16011.69379	339	0.1232 ± -0.0084	0.0718 ± 0.1227	-0.0388 ± 0.0121
92	15188.82488	4561.631505	4539	0.1196 ± -0.0119	0.0781 ± 0.129	-0.046 ± 0.0049
93	15199.0615	15414.51477	198	0.119 ± -0.0126	0.0792 ± 0.1301	-0.036 ± 0.0149
94	15217.9253	12160.08048	257	0.1112 ± -0.0204	0.0702 ± 0.1211	-0.0424 ± 0.0085
95	15362.5331	14683.34004	260	0.1169 ± -0.0147	0.0587 ± 0.1096	-0.0173 ± 0.0335
96	15441.6503	13590.03165	339	0.1218 ± -0.0098	0.0682 ± 0.119	-0.0323 ± 0.0185
97	15476.7292	13686.52602	174	0.1224 ± -0.0091	0.0665 ± 0.1174	-0.0645 ± -0.0136
98	15523.48576	15255.86498	192	0.0916 ± -0.04	0.0612 ± 0.1121	-0.0426 ± 0.0083
99	16121.6486	14526.93494	155	0.1186 ± -0.013	0.067 ± 0.1179	-0.044 ± 0.0069
100	16137.6463	14349.62963	140	0.114 ± -0.0176	0.0628 ± 0.1137	-0.0442 ± 0.0067

Event Number	Reference Time (s)	Average Dominant Frequency (Hz)	Maximum Amplitude (counts)	Locations		
				x coordinate (m)	y coordinate (m)	z coordinate (m)
101	16159.7975	14765.14029	181	0.109 ± -0.0226	0.0823 ± 0.1331	-0.0551 ± -0.0042
102	16177.7625	4260.634921	153	0.115 ± -0.0166	0.0651 ± 0.116	-0.0449 ± 0.006
103	16243.6336	4154.62963	141	0.1123 ± -0.0193	0.0715 ± 0.1224	-0.0659 ± -0.015
104	16300.4352	14902.22222	153	0.1372 ± 0.0056	0.0838 ± 0.1346	-0.0468 ± 0.0041
105	16551.7207	15175.92124	202	0.1114 ± -0.0202	0.0702 ± 0.1211	-0.034 ± 0.0169
106	16662.7561	14309.14205	134	0.1047 ± -0.0269	0.0728 ± 0.1237	-0.0592 ± -0.0084
107	16679.2402	14697.74965	145	0.1117 ± -0.0199	0.0592 ± 0.1101	-0.0648 ± -0.0139
108	16729.4277	14962.22222	199	0.1116 ± -0.02	0.0708 ± 0.1216	-0.0397 ± 0.0112
109	16804.6869	12975.02347	324	0.1112 ± -0.0204	0.064 ± 0.1148	0.0074 ± 0.0582
110	16840.8241	15624.16315	332	0.1079 ± -0.0237	0.0723 ± 0.1232	-0.0533 ± -0.0024

G. Locations of the microseismic events

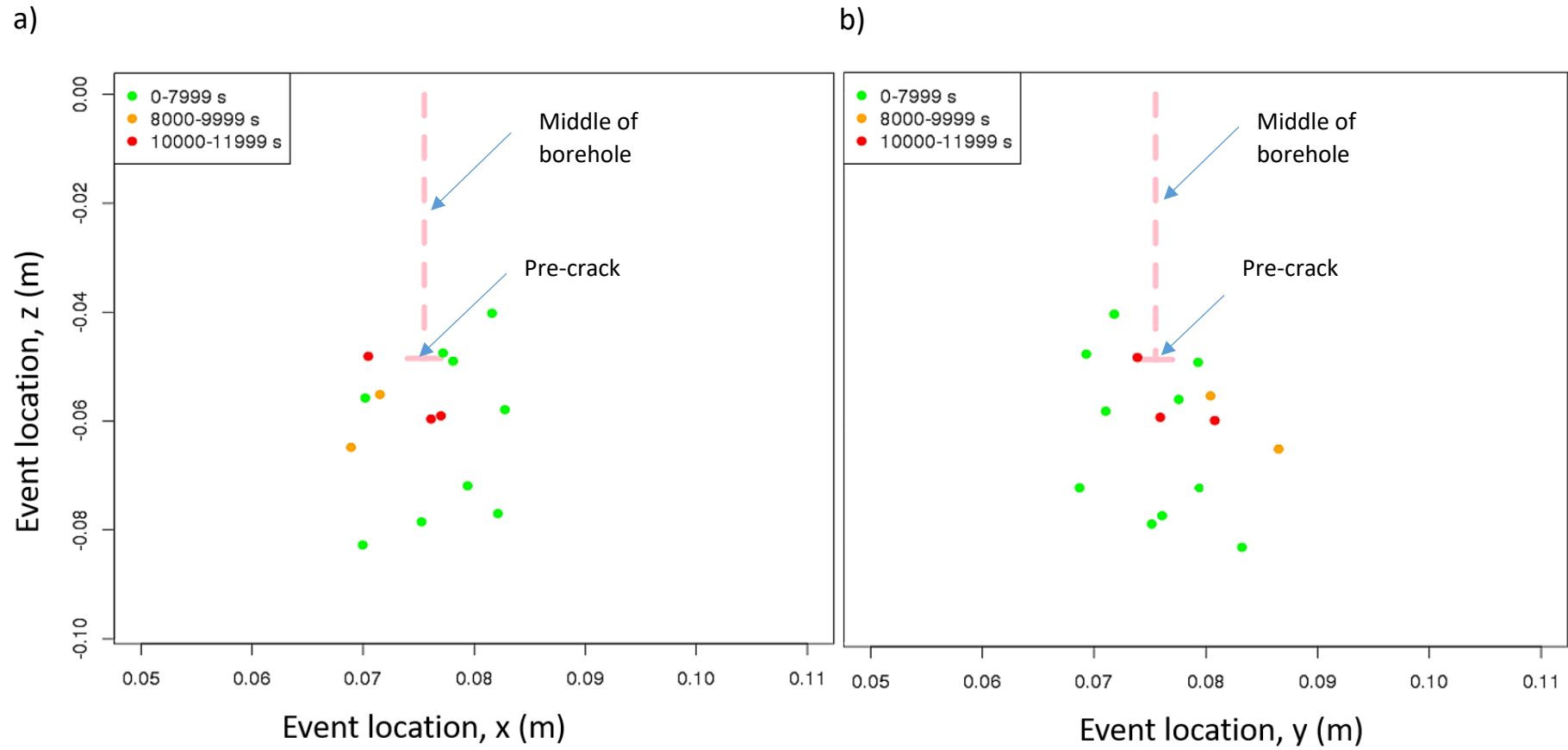


Figure G.1. X, Y and Z locations for the microseismic events occurring in experiment 1. The colors correspond to the timing of the microseismic events. (a) X and Z locations of the microseismic events. (b) Y and Z locations of the microseismic events.

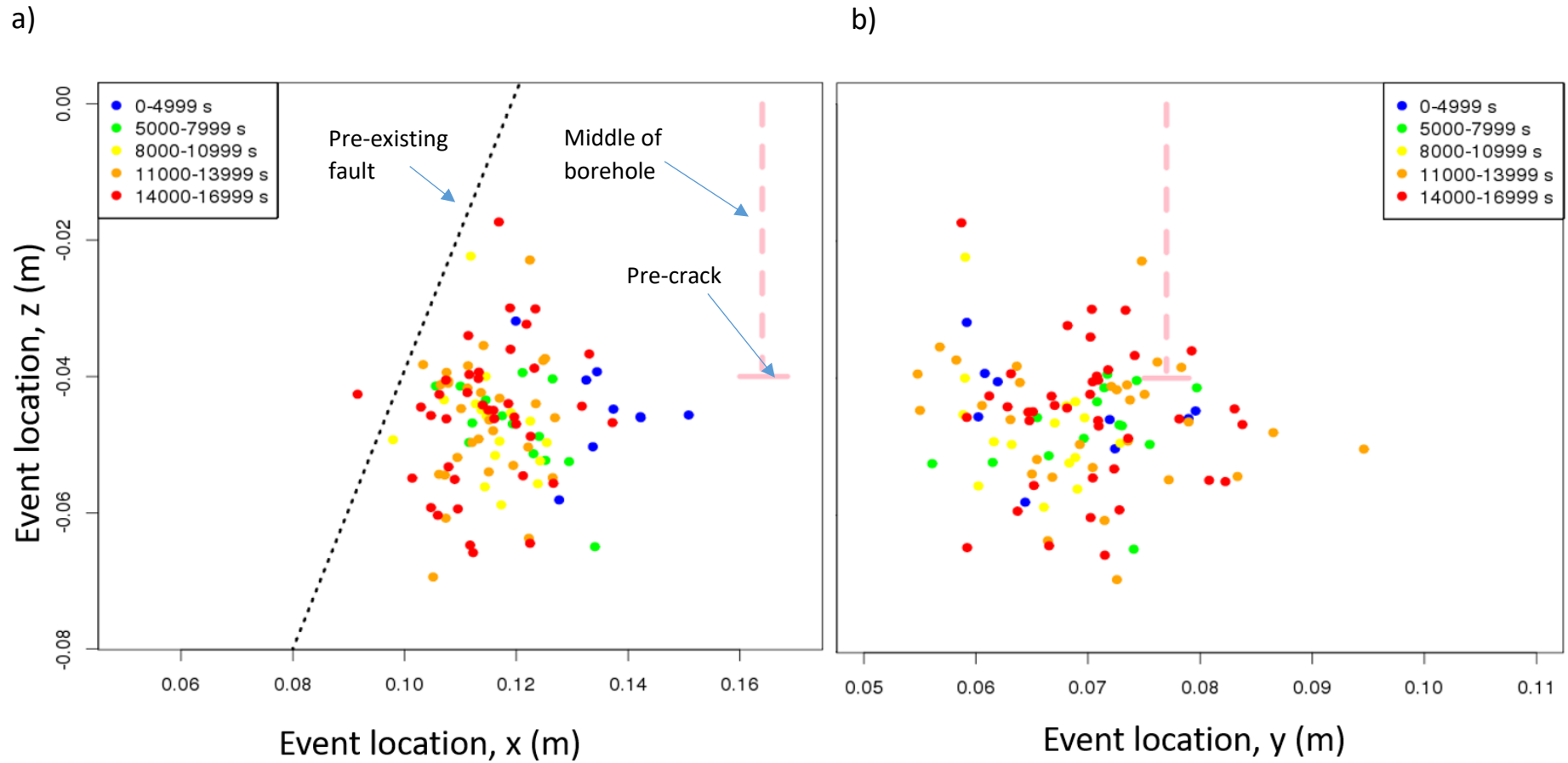


Figure G.2. X, Y and Z locations for the microseismic events occurring in experiment 2. The colors correspond to the timing of the microseismic events. (a) X and Z locations of the microseismic events. (b) Y and Z locations of the microseismic events.

H. Seismic, pump pressure, and camera data in reference time frame Experiment 1

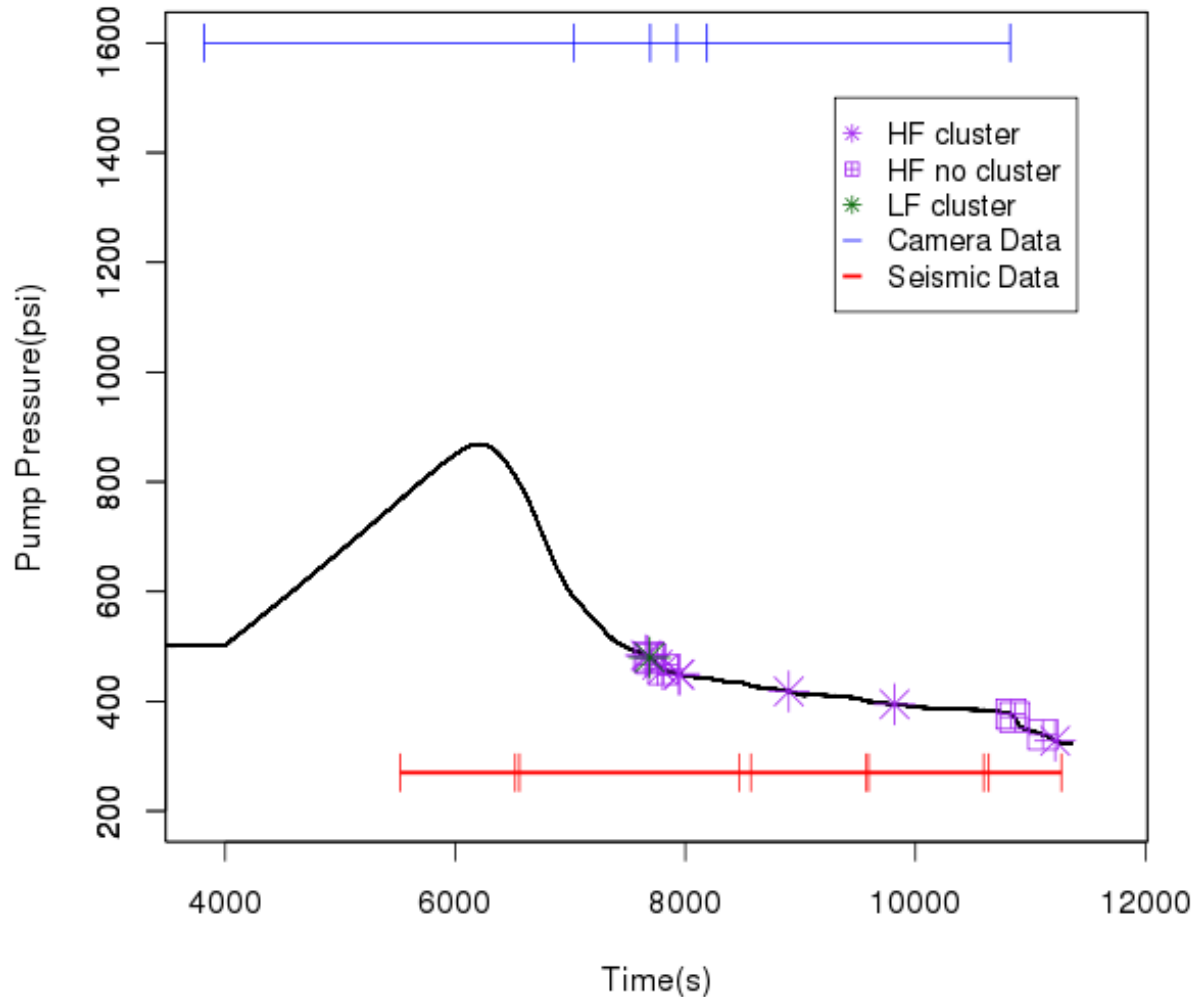


Figure H.1. The pump pressure response during the course of experiment 1 showing where the seismic and camera data were collected relative to the reference time frame for the experiment with time 0 at the start of the experiment. The pressure gradually increases until the breakdown pressure is reached (~ 810 psi), where it levels and then rapidly declines until the pressure reaches approximately 450 psi. The pressure decline is gradual with two main drops in the pressure both ~ 50 psi. Overall the pressure declines to 350 psi at the end of the experiment. Experiment 1 contains three types of events- high frequency cluster, high frequency single event and one low frequency cluster. These events mainly occur at or along the rapid drops in pressure mentioned with only 2 of the 16 identified events occurring on the gentler slope of the decline.

I. Seismic, pump pressure, and camera data in reference time frame Experiment 2

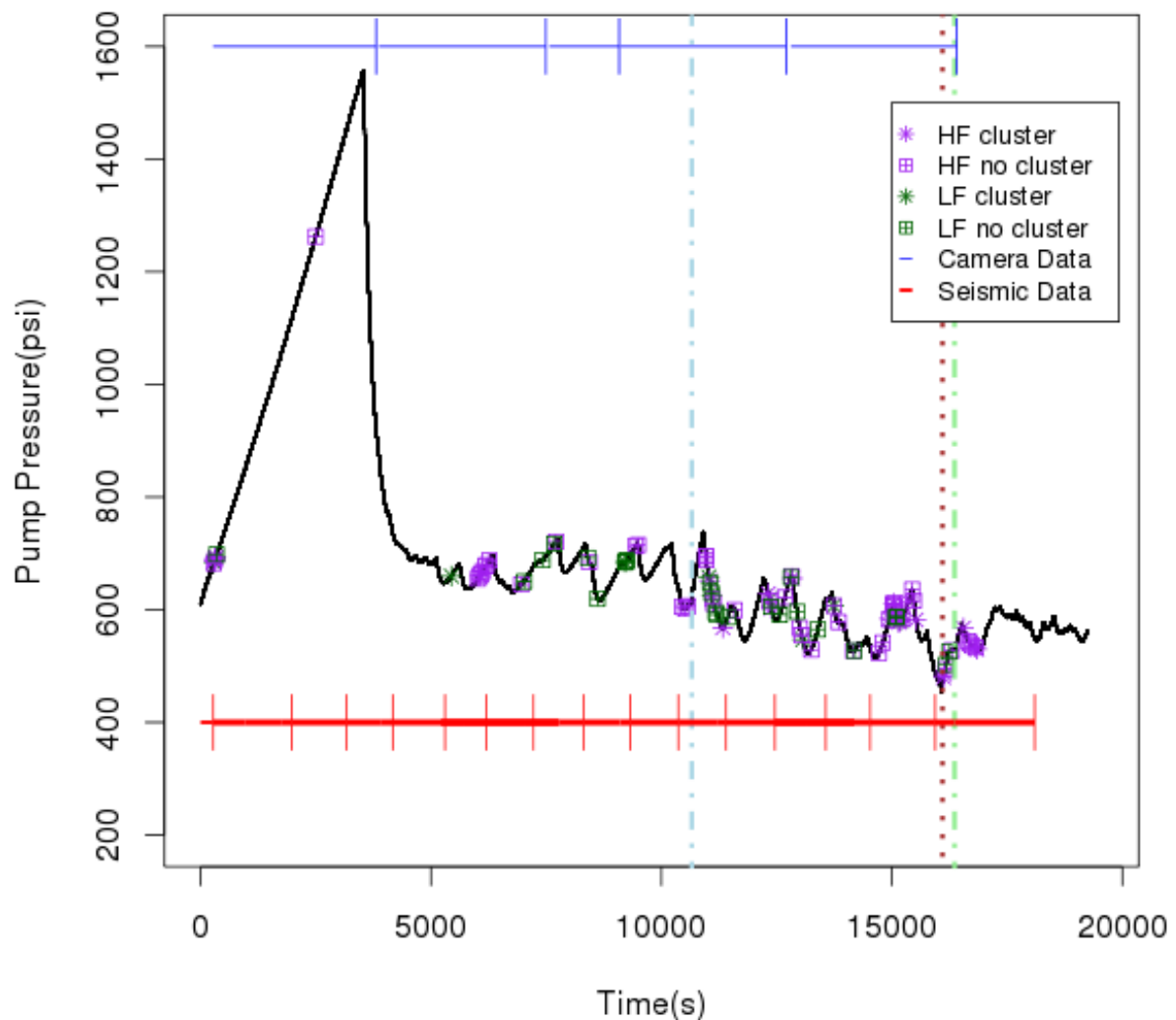


Figure I.1. The pump pressure response during the course of experiment 2 showing where the seismic and camera data were collected relative to the reference time frame for the experiment with time 0 at the start of the experiment. The pressure gradually increases until the breakdown pressure is reached (~ 1600 psi), where it rapidly declines until the pressure reaches approximately 650 psi. The overall pressure decline is gradual after the first main pressure drop, however the pressure oscillates as it declines. Overall, the pressure declines to 600 psi at the end of the experiment but with short increases and decreases. Experiment 2 contains four main types of events- high frequency cluster, high frequency single event and low frequency clusters and low frequency single events. These events mainly occur at or along the increasing and decreasing slopes of the oscillations as well as close to the maxima and minima of the oscillations.

J. Programs

1. Pre-processing.....	129
2. Event Characteristic Processing	130
3. Signal_to_Noise_automated_AM.R :.....	131
4. Frequency_components_auto_AM.R	159
5. Amp_spect_sub.R.....	184
6. Magnitude_Brunemodel_AM.R :	187
7. Location steps- PCA_AM.R.....	212
8. Location steps: Plot_PCA_pract_120914.R.....	246
9. Location steps: plotting_locations_AM.R	309

1. Pre-processing

The following flow-chart shows the pre-processing steps and the programs used. The programs listed here are by Dr. Juan Lorenzo using functions from the R modules RHFM and RSEIS. Parse_Inint_Data.R reads in the raw data, identifies and saves files with a maximum amplitude greater than the threshold set by the user (start with 80 counts as this is the noise level). We use Xcor_t_shift_packet.R and then Xcor_t_shift_event.R to concatenate and correlate the saved files and events from Parse_Inint_Data.R. Xample_GalperinRotate3D.R rotates the 3 components into the principal component system (N, E and Z).

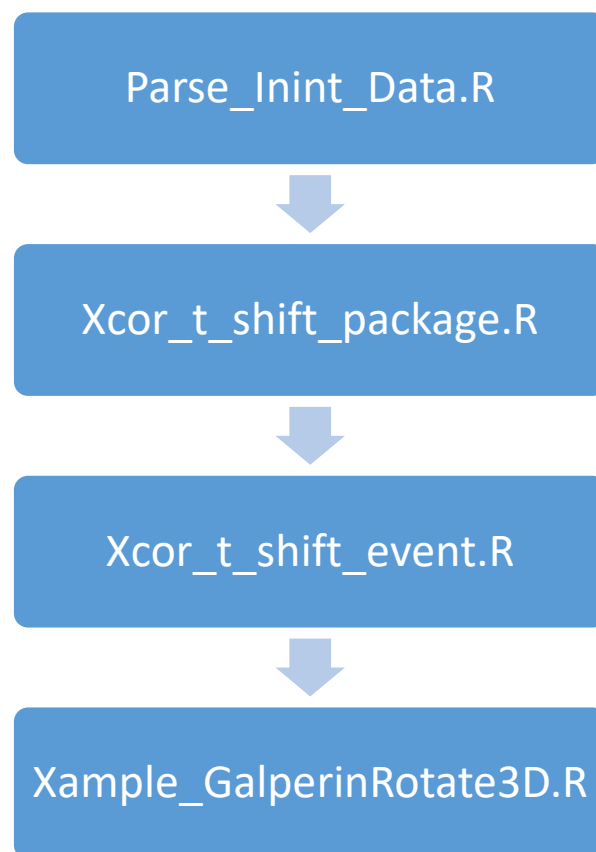


Figure J.1. Flowchart showing the order that the programs should be used when pre-processing the raw data recorded from the piezoelectric sensors.

2. Event Characteristic Processing

The following Rstudio scripts calculate the SNR, frequency, magnitude and locations for the microseismic events picked from the laboratory experiments.

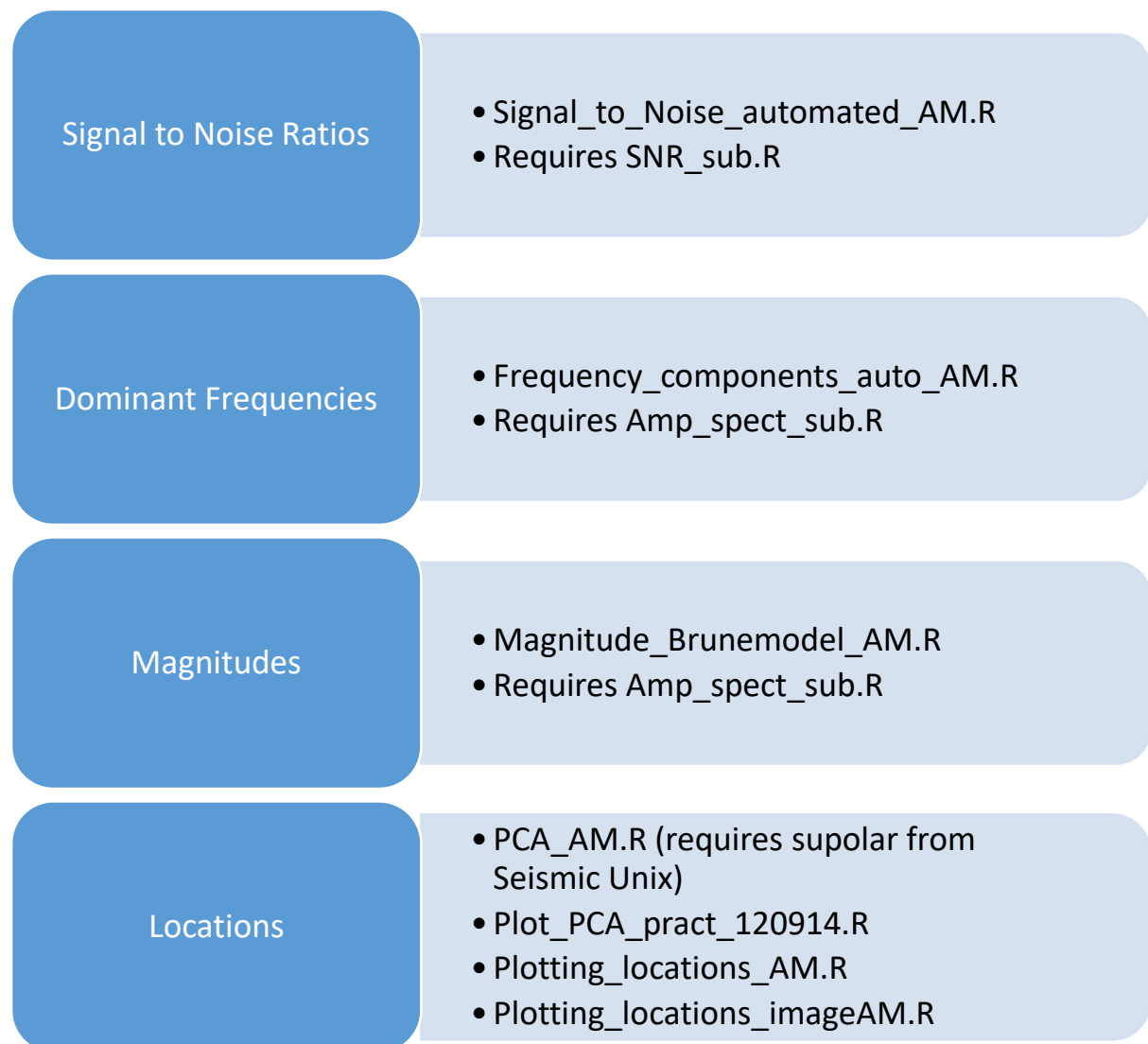


Figure J.2. Flowchart showing the programs that we use to obtain the event characteristics for the categorization of events and mapping of each event type within each sample.

3. Signal_to_Noise_automated_AM.R :

```
#Signal-to-Noise ratio calculation
# SNR = (RMS(Signal)/RMS(Noise))^2
#SNR_dB = 20*log10*(RMS(Signal)/RMS(Noise))
# Abigail Maxwell
#Feb 15th 2016

setwd("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/1")

DIR          <- Project_Dirs("Project_Variables")

data_info <- list()
decimated_stn <- list()
wind_decimated_stn <- list()
noise      <- list()
signal     <- list()
stacked_stn  <- list()

RMS_SIG_X <- list()
RMS_SIG_Y <- list()
RMS_SIG_Z <- list()

RMS_N_X <- list()
RMS_N_Y <- list()
RMS_N_Z <- list()

SNR_X <-list()
SNR_Y <-list()
SNR_Z <-list()
```

```

stacked_signal    <- list()
stacked_noise     <- list()
wind_decimated_stn <- list()

RMS_ST_SIG <- list()
RMS_ST_N   <- list()
SNR_stacked <- list()

#Sample rate (S/s) and sample interval
data_info$sample_rate      = 1.0e6
data_info$sample_int_s     = 1/data_info$sample_rate

Eventlist <- read.delim(paste(DIR$DATA_SEISMIC_TXT, "/Events_starttimes_April16.txt", sep=""))

#event = Eventlist$Event

for(e in 107:111){

print(paste("Event ",e, sep=""))

#Event information
data_info$t_s_start      = Eventlist$File_Start[[e]]
data_info$t_s_end        = Eventlist$File_End[[e]]
data_info$event          = Eventlist$Event[[e]]

#Bin
data_info$order          = Eventlist$Bin[[e]]

```

```
setwd(paste("/home/abbym/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/",data_info$order, sep=""))
```

```
DIR <- Project_Dirs("Project_Variables")
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnA",data_info)
```

```
stnA <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnB",data_info)
```

```
stnB <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnC",data_info)
```

```
stnC <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnD",data_info)
```

```
stnD <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnE",data_info)
```

```
stnE <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnF",data_info)
```

```
stnF <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnG",data_info)
```

```
stnG <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnH",data_info)
```

```
stnH <- data
```

```
list() -> data
```

#Remember after the Xample rotate step, X-N, Y- Up, Z-East

```
STN <- list()
```

```
STN[[1]] <- stnA
```

```
STN[[2]] <- stnB
```

```
STN[[3]] <- stnC
```

```
STN[[4]] <- stnD
```

```
STN[[5]] <- stnE
```

```
STN[[6]] <- stnF
```

```
STN[[7]] <- stnG
```

```
STN[[8]] <- stnH
```

```
data_info$station_nums <- list()
```

```
data_info$station_nums[1] = "A"
```

```
data_info$station_nums[2] = "B"
```

```
data_info$station_nums[3] = "C"
```

```
data_info$station_nums[4] = "D"
```

```
data_info$station_nums[5] = "E"
```

```
data_info$station_nums[6] = "F"
```

```
data_info$station_nums[7] = "G"
```

```
data_info$station_nums[8] = "H"
```

```
#Plot the each component
```

```
for (i in 1:8){
```

```
  decimated_stn[[i]] <- list()
```

```
#Decimate the data to make it easier to plot
```

```
decimation_inc = 1000
```

```
decimated_stn[[i]]$X <- decimate_time_series(STN[[i]]$X,  
      from = data_info$t_s_start,  
      to   = data_info$t_s_end,  
      by   = decimation_inc,  
      with = data_info$sample_int_s)
```

```
decimated_stn[[i]]$Y <- decimate_time_series(STN[[i]]$Y,  
      from = data_info$t_s_start,  
      to   = data_info$t_s_end,  
      by   = decimation_inc,  
      with = data_info$sample_int_s)
```

```
decimated_stn[[i]]$Z <- decimate_time_series(STN[[i]]$Z,  
      from = data_info$t_s_start,  
      to   = data_info$t_s_end,  
      by   = decimation_inc,  
      with = data_info$sample_int_s)
```

```
decimated_t      <- decimate_time_series(NULL,  
      from = data_info$t_s_start,  
      to   = data_info$t_s_end,  
      by   = decimation_inc,  
      with = data_info$sample_int_s)
```

```
if (i %in% 1:4) {
```



```

plot.new()

#4 Stations on one plot
par(mfrow=c(4,1))

#Plot each component on one graph
bias1 = 0
bias2= 600
bias3= 1000

par(mfg=c(i,1))

plot(decimated_t,demean(unlist(decimated_stn[[i]]$X)) + bias1,
     ylim = c(-500, 1500),
     #xlim = PLOT$xlim_raw,
     type = "l",
     col = "red",
     main = "N-red E-green Z-blue stnA",
     xlab = "Time (s)",ylab="Amp (counts)", sub="" )

lines(decimated_t,demean(unlist(decimated_stn[[i]]$Z)) + bias2,
     col = "green")

lines(decimated_t,demean(unlist(decimated_stn[[i]]$Y)) + bias3,
     col = "blue")
}

if (i %in% 5:8) {

plot.new()

```

```
#4 Stations on one plot
```

```
par(mfrow=c(4,1))
```

```
#Plot each component on one graph
```

```
bias1 = 0
```

```
bias2= 600
```

```
bias3= 1000
```

```
par(mfg=c((i-4),1))
```

```
plot(decimated_t,demean(unlist(decimated_stn[[i]]$X)) + bias1,
```

```
ylim = c(-500,1500),
```

```
#xlim = PLOT$xlim_raw,
```

```
type = "l",
```

```
col = "red",
```

```
main = "N-red E-green Z-blue stnA",
```

```
xlab = "time (s)",sub="" )
```

```
lines(decimated_t,demean(unlist(decimated_stn[[i]]$Z)) + bias2,
```

```
col = "green")
```

```
lines(decimated_t,demean(unlist(decimated_stn[[i]]$Y)) + bias3,
```

```
col = "blue")
```

```
}
```

```
}
```

```
#Calculation of signal to noise ratio
```

```
for (i in 1:1){
```

```
#Choose part of signal to be used
```

```
#dev.off()
```

```

questioning = TRUE

#Choose window
while (questioning) {

  plot.new()

  par(new=FALSE)

  par(mfrow=c(1,1))

  plot(decimated_t,demean(unlist(decimated_stn[[i]]$Y)),
       ylim = c(min(demean(unlist(decimated_stn[[i]]$Y)))-100,
max(demean(unlist(decimated_stn[[i]]$Y))+100),
       #xlim = PLOT$xlim_raw,
       type = "l",
       col = "red",
       main = paste("N-red stn",data_info$station_nums[[i]]),
       xlab = "Time (s)",ylab="Amp (counts)", sub="" )

  ans = readline("Choose a window? ")

  if (ans== "y") {

    print("Choose window by selecting two points")

    wind_points <- identify(decimated_t,demean(unlist(decimated_stn[[i]]$Y)),n=2,
pos=FALSE,plot=FALSE)

    #wind_points <- locator(n=2, type="n")

```

```

wind_start_x <- wind_points[[1]]
wind_end_x <- wind_points[[2]]
num_samples_wind <- wind_end_x - wind_start_x

print(paste("Num of samples: ", num_samples_wind, sep=""))

#Signal window

signal[[i]] <- list()

signal[[i]]$X <- decimated_stn[[i]]$X[wind_start_x:wind_end_x]
signal[[i]]$Y <- decimated_stn[[i]]$Y[wind_start_x:wind_end_x]
signal[[i]]$Z <- decimated_stn[[i]]$Z[wind_start_x:wind_end_x]

#Noise Window

noise[[i]] <- list()

noise[[i]]$X <- decimated_stn[[i]]$X[1:length(signal[[i]]$X)]
noise[[i]]$Y <- decimated_stn[[i]]$Y[1:length(signal[[i]]$Y)]
noise[[i]]$Z <- decimated_stn[[i]]$Z[1:length(signal[[i]]$Z)]

# decimation_inc = 1
#
wind_decimated_stn[[i]] <- list()

wind_decimated_stn[[i]]$Y <- decimated_stn[[i]]$Y[wind_start_x:wind_end_x]

# wind_decimated_stn[[i]]$Y <- decimate_time_series(signal[[i]]$Y,

```

```

#                               from = wind_start_x*data_info$sample_int + 1,
#                               to  = wind_end_x*data_info$sample_int + 1,
#                               by   = decimation_inc,
#                               with = data_info$sample_int_s)
#
# wind_decimated_stn[[i]]$Z <- decimate_time_series(signal[[i]]$Z,
#                               from = wind_start_x*data_info$sample_int + 1,
#                               to  = wind_end_x*data_info$sample_int + 1,
#                               by   = decimation_inc,
#                               with = data_info$sample_int_s)

wind_decimated_t  <- seq(from=decimated_t[[wind_start_x]], to=decimated_t[[wind_end_x]],
length= length(signal[[i]]$Y))

# plot(decimated_t,demean(unlist(decimated_stn[[i]]$X)),
#      ylim = c(min(demean(unlist(decimated_stn[[i]]$X)))-100,
#                max(demean(unlist(decimated_stn[[i]]$X))+100),
#      #xlim = PLOT$xlim_raw,
#      type = "l",
#      col = "red",
#      main = "N-red stnA",
#      xlab = "Time (s)",ylab="Amp (counts)", sub="" )

pick1_x <- decimated_t[[wind_start_x]]
pick1_y <- decimated_stn[[i]]$Y[[wind_start_x]]

pick2_x <- decimated_t[[wind_end_x]]
pick2_y <- decimated_stn[[i]]$Y[[wind_end_x]]

PLTpicks(pick1_x,col="black")
PLTpicks(pick2_x,col="black")

```

```

noise_pick1_x <- decimated_t[[1]]
noise_pick1_y <- decimated_stn[[i]]$Y[[1]]

noise_pick2_x <- decimated_t[[length(noise[[i]]$Y)]]
noise_pick2_y <- decimated_stn[[i]]$Y[[length(noise[[i]]$Y)]]

PLTpicks(noise_pick1_x,col="blue")
PLTpicks(noise_pick2_x,col="blue")

plot.new()

par(new=FALSE)

par(mfrow=c(1,1))

plot(wind_decimated_t,demean(unlist(wind_decimated_stn[[i]]$Y)),
      ylim = c(min(demean(unlist(wind_decimated_stn[[i]]$Y)))-100,
max(demean(unlist(wind_decimated_stn[[i]]$Y))+100),
      #xlim = PLOT$xlim_raw,
      type = "l",
      col = "red",
      main = paste("STN",data_info$station_nums[[i]], sep=""),
      xlab = "Time (s)",ylab="Amp (counts)", sub="" )

```

```

ans = readline("Is this window correct? ")

if(substr(ans,1,1) == "y") {
questioning = FALSE

}
else {
questioning = TRUE
}

}

else
{}
}
}

for (i in 2:8){
#Signal window

signal[[i]] <- list()

signal[[i]]$X <- decimated_stn[[i]]$X[wind_start_x:wind_end_x]
signal[[i]]$Y <- decimated_stn[[i]]$Y[wind_start_x:wind_end_x]
signal[[i]]$Z <- decimated_stn[[i]]$Z[wind_start_x:wind_end_x]

#Noise Window

noise[[i]] <- list()

noise[[i]]$X <- decimated_stn[[i]]$X[1:length(signal[[i]]$X)]

```

```

noise[[i]]$Y <- decimated_stn[[i]]$Y[1:length(signal[[i]]$Y)]
noise[[i]]$Z <- decimated_stn[[i]]$Z[1:length(signal[[i]]$Z)]

# decimation_inc = 1
#
wind_decimated_stn[[i]] <- list()

wind_decimated_stn[[i]]$Y <- decimated_stn[[i]]$Y[wind_start_x:wind_end_x]

# wind_decimated_stn[[i]]$Y <- decimate_time_series(signal[[i]]$Y,
#
#           from = wind_start_x*data_info$sample_int + 1,
#           to  = wind_end_x*data_info$sample_int + 1,
#           by  = decimation_inc,
#           with = data_info$sample_int_s)
#
# wind_decimated_stn[[i]]$Z <- decimate_time_series(signal[[i]]$Z,
#
#           from = wind_start_x*data_info$sample_int + 1,
#           to  = wind_end_x*data_info$sample_int + 1,
#           by  = decimation_inc,
#           with = data_info$sample_int_s)

wind_decimated_t  <- seq(from=decimated_t[[wind_start_x]], to=decimated_t[[wind_end_x]],
length= length(signal[[i]]$Y))

# plot(decimated_t,demean(unlist(decimated_stn[[i]]$X)),
#   ylim = c(min(demean(unlist(decimated_stn[[i]]$X)))-100,
# max(demean(unlist(decimated_stn[[i]]$X))+100),
#   #xlim = PLOT$xlim_raw,
#   type = "l",
#   col = "red",
#   main = "N-red stnA",
#   xlab = "Time (s)",ylab="Amp (counts)", sub="" )

```



```

pick1_x <- decimated_t[[wind_start_x]]
pick1_y <- decimated_stn[[i]]$Y[[wind_start_x]]

pick2_x <- decimated_t[[wind_end_x]]
pick2_y <- decimated_stn[[i]]$Y[[wind_end_x]]

PLTpicks(pick1_x,col="black")
PLTpicks(pick2_x,col="black")

noise_pick1_x <- decimated_t[[1]]
noise_pick1_y <- decimated_stn[[i]]$Y[[1]]

noise_pick2_x <- decimated_t[[length(noise[[i]]$Y)]]
noise_pick2_y <- decimated_stn[[i]]$Y[[length(noise[[i]]$Y)]]

PLTpicks(noise_pick1_x ,col="blue")
PLTpicks(noise_pick2_x ,col="blue")

plot.new()

par(new=FALSE)

par(mfrow=c(1,1))

plot(wind_decimated_t,demean(unlist(wind_decimated_stn[[i]]$Y)),

```

```

ylim = c(min(demean(unlist(wind_decimated_stn[[i]]$Y))-100,
max(demean(unlist(wind_decimated_stn[[i]]$Y))+100),

#xlim = PLOT$xlim_raw,

type = "l",

col = "red",

main = paste("STN",data_info$station_nums[[i]], sep=""),

xlab = "Time (s)",ylab="Amp (counts)", sub="" )

}

```

```

for( i in 1:8){

```

```

#Calculation of S/N for each component for each individual station
print("Calculation of SNR for each component of each station")

```

```

RMS_SIG_X[[i]] <- list()
RMS_SIG_Y[[i]] <- list()
RMS_SIG_Z[[i]] <- list()

```

```

RMS_N_X[[i]] <- list()
RMS_N_Y[[i]] <- list()
RMS_N_Z[[i]] <- list()

```

```

SNR_X[[i]] <-list()
SNR_Y[[i]] <-list()
SNR_Z[[i]] <-list()

```

```

RMS_SIG_X[[i]] <- sqrt(mean((signal[[i]]$X)^2))
RMS_SIG_Y[[i]] <- sqrt(mean((signal[[i]]$Y)^2))
RMS_SIG_Z[[i]] <- sqrt(mean((signal[[i]]$Z)^2))

```

```

RMS_N_X[[i]] <- sqrt(mean((noise[[i]]$X)^2))

```

```
RMS_N_Y[[i]] <- sqrt(mean((noise[[i]]$Y)^2))
```

```
RMS_N_Z[[i]] <- sqrt(mean((noise[[i]]$Z)^2))
```

```
#SNR = (RMS(Signal)/RMS(Noise))^2
```

```
SNR_X[[i]] <- (RMS_SIG_X[[i]]/RMS_N_X[[i]])^2
```

```
SNR_Y[[i]] <- (RMS_SIG_Y[[i]]/RMS_N_Y[[i]])^2
```

```
SNR_Z[[i]] <- (RMS_SIG_Z[[i]]/RMS_N_Z[[i]])^2
```

```
#STACKED STATIONS
```

```
print("Stacking X, Y, Z components of each station for average SNR to be calculated")
```

```
su_data_out <- list()
```

```
#Calculation of S/N for each station (with stacked components) for an event
```

```
su_data_out$num_samples = length(decimated_stn[[i]]$X)
```

```
startY = 1
```

```
endY = su_data_out$num_samples
```

```
startX = su_data_out$num_samples+1
```

```
endX = 2*su_data_out$num_samples
```

```
startZ = 2*su_data_out$num_samples +1
```

```
endZ = 3*su_data_out$num_samples
```

```
data_length = 3 * as.numeric(su_data_out$num_samples)
```

```
su_data_out_vector = vector(mode="numeric",length=data_length)
```

```

su_data_out_vector[startY:endY] <- decimated_stn[[i]]$Y
su_data_out_vector[startX:endX] <- decimated_stn[[i]]$X
su_data_out_vector[startZ:endZ] <- decimated_stn[[i]]$Z

stn = data_info$station_nums[i]

binfile      = paste("order",data_info$order,"_event",data_info$event,"_STN", stn,".bin",sep="")
outbound_bin  = file.path(DIR$DATA_SEISMIC_SU,binfile)
outbound_connection = file(outbound_bin,"wb")

# output data to su binary file
writeSuBinData(su_data_out_vector,outbound_connection)
close(outbound_connection)
print(paste("Wrote ",binfile,sep=""))

binfile      = paste("order",data_info$order,"_event",data_info$event,"_STN", stn,sep="")

#system(paste("cd ",sep="",DIR$PL_SEISMIC,";",
#"perl Supolar_"))

#Remember after the Xample rotate step, X-N, Y- Up, Z-East
setwd(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/",data_info$order,"/su", sep=""))

system(paste("suaddhead ns=",su_data_out$num_samples,"<", binfile,".bin", " | sushw key=dt
a=100 | sustack >",binfile,".su",sep=""))

#Read in stacked signal for SNR
sufile      = paste(binfile,".su",sep="")
num_last_trace = 1
num_traces = 1
num_samples_per_trace = su_data_out$num_samples

```

```

trace_nums      = c(1:num_last_trace)
num_samples     = num_traces * num_samples_per_trace
inbound         = paste(DIR$DATA_SEISMIC_SU,
                        sufile,sep="/"
)
connection      <- file(inbound,"rb")
#for (i in trace_nums) {

stacked_stn[[i]] <- list()

stacked_stn[[i]] = readSu (connection,num_samples)

close(connection)

decimated_stacked_t <- seq(from= data_info$t_s_start, to=data_info$t_s_end,
length=su_data_out$num_samples)

#Choose part of signal to be used
#dev.off()

#Window same as above

plot.new()

par(new=FALSE)

par(mfrow=c(1,1))

```

```

plot(decimated_stacked_t,demean(unlist(stacked_stn[[i]]$data[[1]])),
     ylim = c(min(demean(unlist(stacked_stn[[i]]$data[[1]])))-100,
max(demean(unlist(stacked_stn[[i]]$data[[1]]))+100),
     #xlim = PLOT$xlim_raw,
     type = "l",
     col = "red",
     main = paste("Stacked", data_info$station_nums[[i]]),
     xlab = "Time (s)",ylab="Amp (counts)", sub="" )

```

#Windowing data using same window as chosen above

```

wind_start_x <- wind_points[[1]]
wind_end_x  <- wind_points[[2]]

```

```

num_samples_wind <- wind_end_x - wind_start_x

```

```

print(paste("Num of samples: ", num_samples_wind, sep=""))

```

#Signal window

```

stacked_signal[[i]] <- list()

```

```

stacked_signal[[i]] <- stacked_stn[[i]]$data[[1]][wind_start_x:wind_end_x]

```

#Noise Window

```

stacked_noise[[i]] <- list()

```

```

stacked_noise[[i]] <- stacked_stn[[i]]$data[[1]][1:length(stacked_signal[[i]])]

```

```
decimation_inc <- 1
```

```
wind_decimated_stacked_stn[[i]] <- list()
```

```
wind_decimated_stacked_stn[[i]] <- decimate_time_series(stacked_signal[[i]],  
  from = (wind_start_x*data_info$sample_int/1000) + 1,  
  to   = (wind_end_x*data_info$sample_int/1000) + 1,  
  by   = decimation_inc,  
  with = data_info$sample_int_s/1000)
```

```
wind_decimated_stacked_t  <- decimate_time_series(NULL,  
  from = (wind_start_x*data_info$sample_int/1000) + 1,  
  to   = (wind_end_x*data_info$sample_int/1000) + 1,  
  by   = decimation_inc,  
  with = data_info$sample_int_s/1000)
```

```
pick1_x <- decimated_stacked_t[[wind_start_x]]  
pick1_y <- stacked_stn[[i]]$data[[1]][[wind_start_x]]
```

```
pick2_x <- decimated_stacked_t[[wind_end_x]]  
pick2_y <- stacked_stn[[i]]$data[[1]][[wind_end_x]]
```

```
PLTpicks(pick1_x,col="black")
```

```
PLTpicks(pick2_x,col="black")
```

```
noise_pick1_x <- decimated_stacked_t[[1]]
```

```

noise_pick1_y <- stacked_stn[[i]]$data[[1]][[1]]

noise_pick2_x <- decimated_stacked_t[[length(stacked_noise[[i]])]]
noise_pick2_y <- stacked_stn[[i]]$data[[1]][[length(stacked_noise[[i]])]]

PLTpicks(noise_pick1_x,col="blue")
PLTpicks(noise_pick2_x,col="blue")


plot.new()

par(new=FALSE)

par(mfrow=c(1,1))

plot(wind_decimated_stacked_t,demean(unlist(wind_decimated_stacked_stn[[i]])),
      ylim = c(min(demean(unlist(wind_decimated_stacked_stn[[i]])))-100,
max(demean(unlist(wind_decimated_stacked_stn[[i]])))+100),
      #xlim = PLOT$xlim_raw,
      type = "l",
      col = "red",
      main = paste("Stacked STN", stn, sep=""),
      xlab = "Time (s)",ylab="Amp (counts)", sub="" )

```



```

print("Calculation of the stacked SNR")

#Calculation of S/N for each station stacked
RMS_ST_SIG[[i]] <- list()
RMS_ST_N[[i]] <- list()
SNR_stacked[[i]] <- list()

RMS_ST_SIG[[i]] <- sqrt(mean((stacked_signal[[i]]^2))

RMS_ST_N[[i]] <- sqrt(mean((stacked_noise[[i]]^2))

#SNR = (RMS(Signal)/RMS(Noise))^2
SNR_stacked[[i]] <- (RMS_ST_SIG[[i]]/RMS_ST_N[[i]])^2

setwd(paste("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/",data_info$order
, sep=""))
}

#save SNR values
print ("Saving SNR values of event to table")

SNR_event <- cbind(data_info$station_nums,SNR_X, SNR_Y, SNR_Z,SNR_stacked)

colnames(SNR_event) <- c("Station","SNR_X","SNR_Y","SNR_Z", "SNR_stacked")

SNR_event_outbound <-
paste(DIR$DATA_SEISMIC_TXT_SNR,"/event",data_info$event,"_SNR.txt",sep="")
write.table(SNR_event, file= SNR_event_outbound, sep=" ", col.names=TRUE, row.names=FALSE)
}

```

The following is the sub-script for Signal_to_Noise_automated_AM.R :

```
#SNR subroutine
```

```
#Abigail Maxwell
```

```
#Feb 17th 2016
```

```
#Version 1
```

```
SNR_sub <- function(t, signal, decimate, start, end, sample_int) {
```

```
  questioning = TRUE
```

```
  #Choose window
```

```
  while (questioning) {
```

```
    plot.new()
```

```
    par(new=FALSE)
```

```
    decimation_inc <- decimate
```

```
    data_info$t_s_start <- start
```

```
    data_info$t_s_end <- end
```

```
    data_info$sample_int_s <- sample_int
```

```
    decimated_signal      <- decimate_time_series(signal,
                                                    from = data_info$t_s_start,
                                                    to  = data_info$t_s_end,
                                                    by  = decimation_inc,
                                                    with = data_info$sample_int_s)
```

```
    decimated_t          <- decimate_time_series(NULL,
                                                    from = data_info$t_s_start,
```

```

        to = data_info$t_s_end,
        by = decimation_inc,
        with = data_info$sample_int_s)

par(mfrow=c(1,1))

plot(decimated_t,demean(unlist(decimated_signal)),
      ylim = c(min(demean(unlist(decimated_signal)))-50,
max(demean(unlist(decimated_signal))+50),
      xlim = c(min(decimated_t),max(decimated_t)),
      type = "l",
      col = "red",
      main = "Windowed signal",
      xlab = "Time (s)",ylab="Amp", sub="" )

ans = readline("Choose a window? ")

if (ans== "y") {

  print("Choose window by selecting two points")

  wind_points <- identify(decimated_t,demean(unlist(decimated_signal)),n=2,
pos=FALSE,plot=FALSE)

  wind_start_x <- wind_points[[1]]
  wind_end_x <- wind_points[[2]]

  #Signal window

  signal_wind <- list()

```

```
signal_wind <- decimated_signal[wind_start_x:wind_end_x]
```

```
#Noise Window
```

```
noise_wind <- list()
```

```
noise_wind <- decimated_signal[1:length(signal_wind)]
```

```
wind_t <- seq(from=t[[wind_start_x]], to=t[[wind_end_x]], length= length(signal_wind))
```

```
#Plot picks
```

```
par(mfrow=c(1,1))
```

```
plot(t,demean(unlist(decimated_signal)),
```

```
  ylim = c(min(demean(unlist(decimated_signal)))-50,  
max(demean(unlist(decimated_signal))+50),
```

```
  xlim = c(min(t),max(t)),
```

```
  type = "l",
```

```
  col = "red",
```

```
  main = "Windowed signal",
```

```
  xlab = "Time (s)",ylab="Amp", sub="" )
```

```
pick1_x <- decimated_t[[wind_start_x]]
```

```
pick1_y <- decimated_signal[[wind_start_x]]
```

```
pick2_x <- decimated_t[[wind_end_x]]
pick2_y <- decimated_signal[[wind_end_x]]
```

```
PLTpicks(pick1_x,col="black")
PLTpicks(pick2_x,col="black")
```

```
noise_pick1_x <- decimated_t[[1]]
noise_pick1_y <- decimated_signal[[1]]
```

```
noise_pick2_x <- decimated_t[[length(noise_wind)]]
noise_pick2_y <- decimated_signal[[length(noise_wind)]]
```

```
PLTpicks(noise_pick1_x,col="blue")
PLTpicks(noise_pick2_x,col="blue")
```

#Plot the chosen Window

```
plot.new()
```

```
plot(wind_t,demean(unlist(signal_wind)),
     ylim = c(min(demean(unlist(signal_wind)))-50, max(demean(unlist(signal_wind)))+50),
     xlim = c(min(wind_t), max(wind_t)),
     type = "l",
     col = "red",
     main = paste("Chosen Window", sep=""),
     xlab = "Time (s)",ylab="Amp (counts)", sub="" )
```

```

ans = readline("Is this window correct? ")

if(substr(ans,1,1) == "y") {
  questioning = FALSE

}
else {
  questioning = TRUE
}

}

else
{}
}

#Calculation of S/N for each component for each individual station

print("Calculation of SNR for each component of each station")

RMS_SIG <- list()

RMS_N <- list()

SNR <-list()

RMS_SIG <- sqrt(mean((signal_wind )^2))

RMS_N<- sqrt(mean((noise_wind)^2))

```

```
#SNR = (RMS(Signal)/RMS(Noise))^2
```

```
SNR<-(RMS_SIG/RMS_N)^2
```

```
print(paste("SNR =", SNR, sep=""))
```

```
return(SNR)
```

```
}
```

4. Frequency_components_auto_AM.R

The following script plots the amplitude spectrums and saves the dominant frequencies for each component of every event chosen

```
#Frequency Analysis
```

```
#Using the data from Frequency spectrum generated to find mean, mode,
```

```
#and Standard Deviation
```

```
#Can be automated for each component of every event
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
list()       -> opt
```

```
list()       -> info
```

```
list()       -> default
```

```
list()       -> PLOT
```

```
list()       -> comp_sub_trim
```

```
list()       -> decimated_stn
```

```
list()       -> comp2plot
```

```
list()       -> COMP
```

```
list()       -> event_SNR
```



```
list()      -> Dom_freq  
  
list()      -> mode_amp  
  
list()      -> median_freq  
  
list()      -> median_amp  
  
list()      -> sd_freq  
  
list()      -> sd_amp  
  
list()      -> data_info  
  
list()      -> Freq_event  
  
list()      -> trim_data  
  
list()      -> decimated_amp2plot
```

```
opt$freq_statistics <- TRUE
```

```
opt$read           = T #TRUE
```

```
# immediately follows read
```

```
opt$select_stn_ch   = T
```

```
opt$plot_raw_data   = T
```

opt\$plot_spectrogram = T

opt\$mean = T

opt\$mode = T

opt\$median = T

opt\$sd = T

opt\$Brune_analysis = T

opt\$filter = F

opt\$save_data = T

#order = 1

```

Eventlist <-
read.delim(paste("/home/abbym/hydraulicfracturing/seismics/data/HRKE122/AD_card/sen
sor/120914/1/txt/Events_starttimes_April16.txt", sep=""))

for(e in 1:1){

# for(e in 1:1){

# for(e in 37:37){

bin = Eventlist$Bin[[e]]

event = Eventlist$Event[[e]]

#Bin

data_info$order      = bin

```

```
#Sample rate (S/s) and sample interval
```

```
data_info$sample_rate      = 1.0e6
```

```
data_info$sample_int_s     = 1/data_info$sample_rate
```

```
#Event information
```

```
data_info$t_s_start        = Eventlist$File_Start[[e]]
```

```
data_info$t_s_end          = Eventlist$File_End[[e]]
```

```
data_info$event            = event
```

```
setwd(paste("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/",data_info$order, sep=""))
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
#Read in files
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnA",data_info)
```

```
stnA <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnB",data_info)
```

```
stnB <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnC",data_info)
```

```
stnC <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnD",data_info)
```

```
stnD <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnE",data_info)
```

```
stnE <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnF",data_info)
```

```
stnF <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnG",data_info)
```

```
stnG <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnH",data_info)
```

```
stnH <- data
```

```
list() -> data
```

```
#Remember after the Xample rotate step, X-N, Y- Up, Z-East
```

```
STN <- list()
```

```
STN[[1]] = stnA
```

```
STN[[2]] = stnB
```

```
STN[[3]] = stnC
```

```
STN[[4]] = stnD
```

```
STN[[5]] = stnE
```

```
STN[[6]] = stnF
```

```
STN[[7]] = stnG
```

```
STN[[8]] = stnH
```

```
COMP[[1]] <- stnA$Y
```

```
COMP[[2]] <- stnA$X
```

```
COMP[[3]] <- stnA$Z
```

COMP[[4]] <- stnB\$Y

COMP[[5]] <- stnB\$X

COMP[[6]] <- stnB\$Z

COMP[[7]] <- stnC\$Y

COMP[[8]] <- stnC\$X

COMP[[9]] <- stnC\$Z

COMP[[10]] <- stnD\$Y

COMP[[11]] <- stnD\$X

COMP[[12]] <- stnD\$Z

COMP[[13]] <- stnE\$Y

COMP[[14]] <- stnE\$X

COMP[[15]] <- stnE\$Z

COMP[[16]] <- stnF\$Y

COMP[[17]] <- stnF\$X

```
COMP[[18]] <- stnF$Z
```

```
COMP[[19]] <- stnG$Y
```

```
COMP[[20]] <- stnG$X
```

```
COMP[[21]] <- stnG$Z
```

```
COMP[[22]] <- stnH$Y
```

```
COMP[[23]] <- stnH$X
```

```
COMP[[24]] <- stnH$Z
```

```
data_info$station_nums <- list()
```

```
data_info$station_nums[1] = "A_Z"
```

```
data_info$station_nums[2] = "A_Y"
```

```
data_info$station_nums[3] = "A_X"
```

```
data_info$station_nums[4] = "B_Z"
```

```
data_info$station_nums[5] = "B_Y"
```

```
data_info$station_nums[6] = "B_X"
```

```
data_info$station_nums[7] = "C_Z"
```



```
data_info$station_nums[8]      = "C_Y"

data_info$station_nums[9]      = "C_X"

data_info$station_nums[10]     = "D_Z"

data_info$station_nums[11]     = "D_Y"

data_info$station_nums[12]     = "D_X"

data_info$station_nums[13]     = "E_Z"

data_info$station_nums[14]     = "E_Y"

data_info$station_nums[15]     = "E_X"

data_info$station_nums[16]     = "F_Z"

data_info$station_nums[17]     = "F_Y"

data_info$station_nums[18]     = "F_X"

data_info$station_nums[19]     = "G_Z"

data_info$station_nums[20]     = "G_Y"

data_info$station_nums[21]     = "G_X"

data_info$station_nums[22]     = "H_Z"

data_info$station_nums[23]     = "H_Y"

data_info$station_nums[24]     = "H_X"
```

```
#Plot the each component
```

```

#Station with best SNR is used

# event_SNR[[e]] <-
read.table(paste(DIR$DATA_SEISMIC_TXT_SNR,"/event",data_info$event,"_events.txt",
sep=""), header=TRUE)

#

# i <- which.max(event_SNR[[e]]$SNR_stacked)

#StnE has best S/N generally for experiment

for (i in 1:1){

stn <- data_info$station_nums[i]

decimated_stn[[i]] <- list()

#Decimate the data to make it easier to plot

```

```
decimation_inc = 1000
```

```
decimated_stn[[i]] <- decimate_time_series(COMP[[i]],  
  
      from = data_info$t_s_start,  
  
      to  = data_info$t_s_end,  
  
      by  = decimation_inc,  
  
      with = data_info$sample_int_s)
```

```
decimated_t      <- decimate_time_series(NULL,  
  
      from = data_info$t_s_start,  
  
      to  = data_info$t_s_end,  
  
      by  = decimation_inc,  
  
      with = data_info$sample_int_s)
```

```
plot(decimated_t,demean(unlist(decimated_stn[[i]])),
```

```
      #ylim = c(-500, 1500),
```

```
#xlim = PLOT$xlim_raw,

type = "l",

col = "red",

main = paste("STN ",stn, sep=""),

xlab = "Time (s)",ylab="Amp (counts)", sub="" )
```

```
questioning = TRUE
```

```
while(questioning){
```

```
trim_data$t_s_start <- data_info$t_s_start
```

```
trim_data$t_s_end <- data_info$t_s_end
```

```
ans = readline_control(paste("Enter new LOCAL T0 (s): [", trim_data$t_s_start,
"]",sep=""))
```

```
if(ans != "") {
```

```
trim_data$t_s_start = as.numeric(ans)
```

```
}
```

```

ans = readline_control(paste("Enter new LOCAL Tlast: [", trim_data$t_s_end, "]", sep=""))

if(ans != "") {

  trim_data$t_s_end = as.numeric(ans)

}

#recalculate the trimmed limits

trim_data$first_sample = trim_data$t_s_start * data_info$sample_rate + 1 -
data_info$t_s_start*data_info$sample_rate

trim_data$last_sample = trim_data$t_s_end * data_info$sample_rate + 1 -
data_info$t_s_start*data_info$sample_rate

trim_data$num_samples = trim_data$last_sample - trim_data$first_sample + 1


# Trim start of component vector

comp_sub_trim[[i]] <- COMP[[i]][trim_data$first_sample : trim_data$last_sample]

```

```

decimated_amp2plot[[i]]      <- decimate_time_series(comp_sub_trim[[i]] ,

                                from = trim_data$t_s_start,

                                to  = trim_data$t_s_end,

                                by  = decimation_inc,

                                with = data_info$sample_int)

```

```

decimated_trim_t              <- decimate_time_series(NULL,

                                from = trim_data$t_s_start,

                                to  = trim_data$t_s_end,

                                by  = decimation_inc,

                                with = data_info$sample_int)

```

```

trim_data$updated_tmin_s      = trim_data$t_s_start

trim_data$updated_tmax_s      = trim_data$t_s_end

PLOT$xlim                     = c(trim_data$updated_tmin_s,
trim_data$updated_tmax_s)

```

```
decimated_amp2plot[[i]] <- demean(unlist(decimated_amp2plot[[i]]))
```

```
plot(decimated_trim_t,decimated_amp2plot[[i]],
```

```
      ylim = c(min(decimated_amp2plot[[i]])-50, max(decimated_amp2plot[[i]])+50),
```

```
      xlim = PLOT$xlim,
```

```
      type = "l",
```

```
      col = "red",
```

```
      main = paste("Trimmed STN", stn),
```

```
      xlab = "time (s)", ylab="Amp (counts)", sub="")
```

```
ans = readline_control("Another trim? y or [n]" )
```

```
if(substr(ans,1,1) == "n") {
```

```
    questioning = FALSE
```

```
} else {
```

```
    questioning = TRUE
```

```
}
```

```
}
```

```
}
```

```
for (i in 1:1){
```

```
  stn <- data_info$station_nums[i]
```

```
  decimated_stn[[i]] <- list()
```

```
  comp_sub_trim[[i]]      <- COMP[[i]][trim_data$first_sample : trim_data$last_sample]
```

```
  print("Amp Spectrum")
```

```
  #info$sample_int_s <- data_info$sample_int_s
```

```
  #spectra <- fft_op(0,500000, comp_sub_trim[[i]], info)
```

```
  spectra <- amp_spect(comp_sub_trim[[i]] , length(comp_sub_trim[[i]] )-1,
```

```
  data_info$sample_rate)
```



```

# freq_outbound <-

paste(DIR$IMAGES_SEISMIC_JPEG_AMPSPEC,"/event",data_info$event,"_STN",
stn,"_ampspec.jpeg",sep="")

#

# jpeg(file=freq_outbound)

#

# plot(spectra$freqs2plot[-1], demean(unlist(spectra$Amplitude[-1])),

      # ylim = c(min(demean(unlist(spectra$Amplitude[-1])))-10,
max(demean(unlist(spectra$Amplitude[-1]))+10),

      #xlim = c(min(spectra$freqs2plot),max(spectra$freqs2plot)),

      #ylim = c(-30,20),

      # type = "l",

      #col = "blue",

      #main = paste("Amplitude Spectrum ",data_info$event, " STN ", stn, sep=""),

      #xlab = "Freq (Hz) (s)",ylab="Amp", sub="" )

#

#Max_Amplitude      = max(demean(unlist(spectra$Amplitude)))

#Max_index          = which(demean(unlist(spectra$Amplitude)) == Max_Amplitude)

```

```

# points((spectra$freqs2plot)[Max_index],demean(unlist(spectra$Amplitude))[Max_index],

#     col = "red",

#     pch = 19)


# dev.off()


dom_freq= paste("Dominant Frequency =",spectra$freqs2plot[Max_index])

print(dom_freq)


# if (opt$freq_statistics){


Dom_freq[[i]] <- spectra$freqs2plot[Max_index]


mode_amp[[i]] <- MODE(spectra$Amplitude)


Mode_amp = paste("Mode Amplitude=",mode_amp[[i]])

print(Mode_amp)


median_freq[[i]] <- median(spectra$freqs2plot)

```

```
Median_freq= paste("Median Frequency=",median_freq[[i]])
```

```
print(Median_freq)
```

```
median_amp[[i]] <- median(spectra$Amplitude)
```

```
Median_amp= paste("Median Amplitude=",median_amp[[i]])
```

```
print(Median_amp)
```

```
sd_freq[[i]] <- sd(spectra$freqs2plot)
```

```
SD_freq= paste("SD Frequency=", sd_freq[[i]])
```

```
print(SD_freq)
```

```

sd_amp[[i]] <- sd(spectra$Amplitude)

SD_amp= paste("SD Amplitude=", sd_amp[[i]])

print(SD_amp)

#}

#Freq_event <- cbind(event,stn,Dom_freq,mode_amp,median_freq, median_amp,sd_freq,
sd_amp)

# colnames(Freq_event) <-
c("Event","Station","Dom_Freq","Mode_Amp","Med_Freq","Med_Amp","SD_Freq",
"SD_Amp")

#

# Freq_event_outbound <-
paste(DIR$DATA_SEISMIC_TXT,"/event",data_info$event,"_FreqStat",sep="")

# write.table(Freq_event, file= Freq_event_outbound, sep=" ", col.names=TRUE,
row.names=FALSE)

#Plot Frequency Spectrum

#ans = readline("Choose a window? ")

```

```

# t <- seq(from=0,length=length(plot_comp), by=1e-3)

#

# plot(t, demean(unlist(plot_comp)),

#   ylim = c(min(demean(unlist(plot_comp)))-10,max(demean(unlist(plot_comp))+10)),

#   #ylim = c(-50,50),

#   #xlim = PLOT$xlim_raw,

#   type = "l",

#   col = "red",

#   main = paste("Original Data ",data_info$event, " STN ", stn, sep=""),

#   xlab = "Time (s)",ylab="Amp (counts)", sub="" )

#

# print("Choose window by selecting two points")

#

# #wind_points <- identify(decimated_t,demean(unlist(plot_comp)),n=2,

pos=FALSE,plot=FALSE)

# wind_points <- identify(t,demean(unlist(plot_comp)),n=2, pos=FALSE,plot=FALSE)

#

# #wind_points <- locator(n=2, type="n")

#

```

```

# wind_start_x <- wind_points[[1]]

# wind_end_x <- wind_points[[2]]

# num_samples_wind <- wind_end_x-wind_start_x

#

# print(paste("Num of samples: ", num_samples_wind, sep=""))

#

# #wind_decimated_t <- seq(from=decimated_t[[wind_start_x]],
# to=decimated_t[[wind_end_x]], length=
# length(demean(unlist(plot_comp))[wind_start_x:wind_end_x]))

#

# wind_decimated_t <- seq(from=t[[wind_start_x]], to=t[[wind_end_x]], length=
# length(demean(unlist(STN[[i]]$X))[wind_start_x:wind_end_x]))

#

# plot.new()

#

# par(new=FALSE)

#

# par(mfrow=c(1,1))

#

```

```

# plot(wind_decimated_t,demean(unlist(plot_comp))[wind_start_x:wind_end_x],

#   ylim = c(min(demean(unlist(plot_comp))[wind_start_x:wind_end_x])-100,
max(demean(unlist(STN[[i]]$X))[wind_start_x:wind_end_x])+100),

#   #xlim = PLOT$xlim_raw,

#   type = "l",

#   col = "red",

#   main = paste("STN",data_info$station_nums[[i]], sep=""),

#   xlab = "Time (s)",ylab="Amp (counts)", sub="" )

#

}

```

```

component <- rep(c("Z","Y","X"), times= 8)

Freq_event[[e]] <- cbind(rep(bin, times= 24),rep(event, times=
24),component,Dom_freq,mode_amp,median_freq,median_amp,sd_freq,sd_amp)

setwd(paste("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/1",
sep=""))

```

```

Freq_event <- as.data.frame(Freq_event[[e]])

colnames(Freq_event) <-
c("Bin", "Event", "Component", "Dom_Freq", "Mode_Amp", "Med_Freq", "Med_Amp", "SD_Fre
q", "SD_Amp")

Freq_event <- as.matrix(Freq_event)

Freq_event_outbound <-
paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/", bin, "/txt/E
vent", e, "_FreqStat.txt", sep="")

write.table(Freq_event, file= Freq_event_outbound, sep=" ", col.names=TRUE,
row.names=FALSE)

}

```


5. Amp_spect_sub.R

The following script is the sub routine required for the frequency analysis.

```
#Frequency analysis
```

```
#based on procedure described in Frequency analysis, FFT, Frequency Spectrum
```

```
#Victoria Uni,by Dr. Michael Sek
```

```
#Author of program Abigail Maxwell
```

```
#17th February, 2016
```

```
#Version 1
```

```
#Amplitude spectrum
```

```
amp_spect <- function(x,num_samples,sample_rate){
```

```
  spectrum <- list()
```

```
  spectra <- list()
```

```
  x <- demean(x)
```

```
  spectrum <- fft(unlist(x))
```

```
index_Nf <- num_samples/2 +1
```

```
spectrum <- spectrum[1:index_Nf]
```

```
spectrum <- spectrum/num_samples
```

```
spectrum[2:length(spectrum)] <- 2*spectrum[2:length(spectrum)]
```

```
spectra$Amplitude <- 20*log10(Mod(spectrum)/max(Mod(spectrum)))
```

```
spectra$df <- sample_rate/num_samples
```

```
spectra$freq2plot <- spectra$df*(seq(from=0, to=(index_Nf-1),  
length=length(spectra$Amplitude)))
```

```
spectra$num_samples <- num_samples
```

```
# plot(spectra$freq2plot, spectra$Amplitude,
```

```
#   ylim = c(min(spectra$Amplitude)-50, max(spectra$Amplitude)+50),
```

```
#   xlim = c(min(spectra$freq2plot),max(spectra$freq2plot)),
```

```
#   type = "l",
```

```
# col = "blue",

# main = "Amplitude Spectrum",

# xlab = "Freq (Hz) (s)",ylab="Amp", sub="" )


return (spectra)


}
```

6. Magnitude_Brunemodel_AM.R :

The following script calculates and saves the magnitudes for the events that occur during the experiment.

```
#Magnitude using Brune's model
```

```
#Using the Frequency spectrum
```

```
#Abigail Maxwell
```

```
#February 2016
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
list()       -> opt
```

```
list()       -> info
```

```
list()       -> default
```

```
list()       -> PLOT
```

```
list()       -> comp_sub_trim
```

```
list()       -> decimated_stn
```

```
list()       -> comp2plot
```

```
list()       -> COMP
```

list()	-> event_SNR
list()	-> Dom_freq
list()	-> mode_amp
list()	-> median_freq
list()	-> median_amp
list()	-> sd_freq
list()	-> sd_amp
list()	-> data_info
list()	-> Freq_event
list()	-> trim_data
list()	-> decimated_amp2plot
# list()	-> Mo
# list()	-> Mo\$orig
# list()	-> Mo\$upper
# list()	-> Mo\$lower
# list()	-> Mw
# list()	-> Mw\$orig
# list()	-> Mw\$upper
# list()	-> Mw\$lower

```
opt$freq_statistics <- TRUE
```

```
opt$read = T #TRUE
```

```
# immediately follows read
```

```
opt$select_stn_ch = T
```

```
opt$plot_raw_data = T
```

```
opt$plot_spectrogram = T
```

```
opt$mean = T
```

```
opt$mode = T
```

```
opt$median = T
```

```
opt$sd = T
```

```
opt$Brune_analysis      = T
```

```
opt$filter              = F
```

```
opt$save_data           = T
```

```
#order                  = 1
```

```
Eventlist <-
```

```
read.delim(paste("/home/abbym/hydraulicfracturing/seismics/data/HRKE122/AD_card/sen  
sor/120914/1/txt/Events_starttimes_April16.txt", sep=""))
```

```
Locations <-
```

```
read.table("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/txt/lo  
c/ALLLocations_withouterrors.txt", header=TRUE, quote="\")
```

```
Loc_error <-
```

```
read.table("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/txt/lo  
c/Loc_errors.txt", header=TRUE, quote="\")
```

```
geo_loc <-
```

```
read.table(paste("/home/abbym/hydraulicfracturing/well/data/HRKE122/AD_card/sensor/1  
20914",1,"Geo_XYZ_edited.txt",sep="/"),header=FALSE, col.names=c("X","Y","Z"))
```

```
stn_X <- geo_loc$X
```

```
stn_Y <- geo_loc$Y
```

```
stn_Z <- geo_loc$Z
```

```
for(e in 104:111){
```

```
  # for(e in 1:1){
```

```
    # for(e in 37:37){
```

```
      bin = Eventlist$Bin[[e]]
```

```
      event = Eventlist$Event[[e]]
```

```
      #Bin
```

```
      data_info$order      = bin
```



```
#Sample rate (S/s) and sample interval
```

```
data_info$sample_rate      = 1.0e6
```

```
data_info$sample_int_s     = 1/data_info$sample_rate
```

```
#Event information
```

```
data_info$t_s_start        = Eventlist$File_Start[[e]]
```

```
data_info$t_s_end          = Eventlist$File_End[[e]]
```

```
data_info$event            = event
```

```
setwd(paste("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/",data_in  
fo$order, sep=""))
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
#Read in files
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnA",data_info)
```

```
stnA <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnB",data_info)
```

```
stnB <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnC",data_info)
```

```
stnC <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnD",data_info)
```

```
stnD <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnE",data_info)
```

```
stnE <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnF",data_info)
```

```
stnF <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnG",data_info)
```

```
stnG <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnH",data_info)
```

```
stnH <- data
```

```
list() -> data
```

```
#Remember after the Xample rotate step, X-N, Y- Up, Z-East
```

```
#stnA$Y is actually Z comp
```

```
#stnA$X is actually Y comp
```

```
#stnA$Z is actually X comp
```

```
STN <- list()
```

```
STN[[1]] = stnA
```

```
STN[[2]] = stnB
```

```
STN[[3]] = stnC
```

```
STN[[4]] = stnD
```

```
STN[[5]] = stnE
```

```
STN[[6]] = stnF
```

STN[[7]] = stnG

STN[[8]] = stnH

COMP[[1]] <- stnA\$Y

COMP[[2]] <- stnA\$X

COMP[[3]] <- stnA\$Z

COMP[[4]] <- stnB\$Y

COMP[[5]] <- stnB\$X

COMP[[6]] <- stnB\$Z

COMP[[7]] <- stnC\$Y

COMP[[8]] <- stnC\$X

COMP[[9]] <- stnC\$Z

COMP[[10]] <- stnD\$Y

COMP[[11]] <- stnD\$X

COMP[[12]] <- stnD\$Z

```
COMP[[13]] <- stnE$Y
```

```
COMP[[14]] <- stnE$X
```

```
COMP[[15]] <- stnE$Z
```

```
COMP[[16]] <- stnF$Y
```

```
COMP[[17]] <- stnF$X
```

```
COMP[[18]] <- stnF$Z
```

```
COMP[[19]] <- stnG$Y
```

```
COMP[[20]] <- stnG$X
```

```
COMP[[21]] <- stnG$Z
```

```
COMP[[22]] <- stnH$Y
```

```
COMP[[23]] <- stnH$X
```

```
COMP[[24]] <- stnH$Z
```

```
data_info$station_nums <- list()
```

```
data_info$station_nums[1] = "A_Z"
```

data_info\$station_nums[2] = "A_Y"
data_info\$station_nums[3] = "A_X"
data_info\$station_nums[4] = "B_Z"
data_info\$station_nums[5] = "B_Y"
data_info\$station_nums[6] = "B_X"
data_info\$station_nums[7] = "C_Z"
data_info\$station_nums[8] = "C_Y"
data_info\$station_nums[9] = "C_X"
data_info\$station_nums[10] = "D_Z"
data_info\$station_nums[11] = "D_Y"
data_info\$station_nums[12] = "D_X"
data_info\$station_nums[13] = "E_Z"
data_info\$station_nums[14] = "E_Y"
data_info\$station_nums[15] = "E_X"
data_info\$station_nums[16] = "F_Z"
data_info\$station_nums[17] = "F_Y"
data_info\$station_nums[18] = "F_X"
data_info\$station_nums[19] = "G_Z"
data_info\$station_nums[20] = "G_Y"

```
data_info$station_nums[21]    = "G_X"
```

```
data_info$station_nums[22]    = "H_Z"
```

```
data_info$station_nums[23]    = "H_Y"
```

```
data_info$station_nums[24]    = "H_X"
```

```
#Plot the each component
```

```
#Station with best SNR is used
```

```
# event_SNR[[e]] <-
```

```
read.table(paste(DIR$DATA_SEISMIC_TXT_SNR,"/event",data_info$event,"_events.txt",  
sep=""), header=TRUE)
```

```
#
```

```
# i <- which.max(event_SNR[[e]]$SNR_stacked)
```

```
#StnE has best S/N generally for experiment
```

```
for (i in 1:1){
```

```
  stn <- data_info$station_nums[i]
```

```
decimated_stn[[i]] <- list()
```

```
#Decimate the data to make it easier to plot
```

```
decimation_inc = 1000
```

```
decimated_stn[[i]] <- decimate_time_series(COMP[[i]],
```

```
      from = data_info$t_s_start,
```

```
      to   = data_info$t_s_end,
```

```
      by   = decimation_inc,
```

```
      with = data_info$sample_int_s)
```

```
decimated_t      <- decimate_time_series(NULL,
```

```
      from = data_info$t_s_start,
```

```
      to   = data_info$t_s_end,
```

```
      by   = decimation_inc,
```



```
with = data_info$sample_int_s)
```

```
plot(decimated_t,demean(unlist(decimated_stn[[i]])),
```

```
  #ylim = c(-500, 1500),
```

```
  #xlim = PLOT$xlim_raw,
```

```
  type = "l",
```

```
  col = "red",
```

```
  main = paste("STN ",stn, sep=""),
```

```
  xlab = "Time (s)",ylab="Amp (counts)", sub="" )
```

```
questioning = TRUE
```

```
while(questioning){
```

```
  trim_data$t_s_start <- data_info$t_s_start
```

```
  trim_data$t_s_end  <- data_info$t_s_end
```

```

ans = readline_control(paste("Enter new LOCAL T0 (s): [", trim_data$t_s_start,
"]",sep=""))

if(ans != "") {

  trim_data$t_s_start = as.numeric(ans)

}

ans = readline_control(paste("Enter new LOCAL Tlast: [", trim_data$t_s_end, "]",sep=""))

if(ans != "") {

  trim_data$t_s_end = as.numeric(ans)

}

#recalculate the trimmed limits

trim_data$first_sample = trim_data$t_s_start * data_info$sample_rate + 1 -
data_info$t_s_start*data_info$sample_rate

trim_data$last_sample = trim_data$t_s_end * data_info$sample_rate + 1 -
data_info$t_s_start*data_info$sample_rate

trim_data$num_samples = trim_data$last_sample - trim_data$first_sample + 1

```

```

# Trim start of component vector

comp_sub_trim[[i]]      <- COMP[[i]][trim_data$first_sample :
trim_data$last_sample]


decimated_amp2plot[[i]]  <- decimate_time_series(comp_sub_trim[[i]] ,

from = trim_data$t_s_start,

to  = trim_data$t_s_end,

by  = decimation_inc,

with = data_info$sample_int)


decimated_trim_t         <- decimate_time_series(NULL,

from = trim_data$t_s_start,

to  = trim_data$t_s_end,

by  = decimation_inc,

with = data_info$sample_int)

```

```

trim_data$updated_tmin_s      = trim_data$t_s_start

trim_data$updated_tmax_s      = trim_data$t_s_end

PLOT$xlim                     = c(trim_data$updated_tmin_s,
trim_data$updated_tmax_s)


decimated_amp2plot[[i]] <- demean(unlist(decimated_amp2plot[[i]]))


plot(decimated_trim_t,decimated_amp2plot[[i]],

      ylim = c(min(decimated_amp2plot[[i]])-50, max(decimated_amp2plot[[i]])+50),

      xlim = PLOT$xlim,

      type = "l",

      col = "red",

      main = paste("Trimmed STN", stn),

      xlab = "time (s)", ylab="Amp (counts)", sub="")


ans = readline_control("Another trim? y or [n]" )

```

```

if(substr(ans,1,1) == "n") {

  questioning = FALSE

} else {

  questioning = TRUE

}

}

}

}

for (i in 1:1){

  stn <- data_info$station_nums[i]

  decimated_stn[[i]] <- list()

  comp_sub_trim[[i]] <- COMP[[i]][trim_data$first_sample : trim_data$last_sample]

```

```

print("Amp Spectrum")

#info$sample_int_s <- data_info$sample_int_s

#spectra <- fft_op(0,500000, comp_sub_trim[[i]], info)

spectra <- amp_spect(comp_sub_trim[[i]] , length(comp_sub_trim[[i]] )-1,
data_info$sample_rate)

# freq_outbound <-
paste(DIR$IMAGES_SEISMIC_JPEG_AMPSPEC,"/event",data_info$event,"_STN",
stn,"_ampspec.jpeg",sep="")

#

# jpeg(file=freq_outbound)

#plot(spectra$freqs2plot[-1], demean(unlist(spectra$Amplitude[-1])),

#ylim = c(min(demean(unlist(spectra$Amplitude[-1])))-10,
max(demean(unlist(spectra$Amplitude[-1]))+10),

#xlim = c(min(spectra$freqs2plot),max(spectra$freqs2plot)),

#ylim = c(-30,20),

#type = "l",

#col = "blue",

```

```

#main = paste("Amplitude Spectrum ",data_info$event, " STN ", stn, sep=""),

#xlab = "Freq (Hz) (s)",ylab="Amp", sub="" )


Max_Amplitude    = max(demean(unlist(spectra$Amplitude)))

Max_index        = which(demean(unlist(spectra$Amplitude)) == Max_Amplitude)


#points((spectra$freqs2plot)[Max_index],demean(unlist(spectra$Amplitude))[Max_index],

        #col = "red",

        #pch = 19)


# dev.off()

dom_freq= paste("Dominant Frequency =",spectra$freqs2plot[Max_index])

print(dom_freq)


#Using best station with best SNR

#STNE ?


#Converting to metres

```

```
raw_data <- list()
```

```
raw_data <- unlist(comp_sub_trim[[i]])
```

```
#Conversion from counts to meters
```

```
data_volts <- raw_data/33e03
```

```
data_volts_gain <- data_volts/4
```

```
Displmnt_m <- data_volts_gain/1e04
```

```
#xc <- get.corner(spectra$freqs2plot[2:spectra$freqs2plot[Max_index]*2],
```

```
demean(unlist(spectra$Amplitude[2:spectra$freqs2plot[Max_index]*2])),data_info$sample  
_int_s, 0.01, 50000, PLOT=TRUE)
```



```
#xc <- get.corner(spectra$freqs2plot,
(spectra$Amplitude[2:spectra$freqs2plot[Max_index]*2]),data_info$sample_int_s, 0.01,
250000, PLOT=TRUE)
```

```
BF <- brune.doom(Displmnt_m ,data_info$sample_int_s, f1=0.01, 100000, PLOTB=TRUE )
```

```
BF1 <- brune.search(spectra$freqs2plot[-1], demean(unlist(spectra$Amplitude[-1])),
0.01,spectra$freqs2plot[Max_index]*2,BF$omega0,BF$corn,BF$tstar0, 1.8)
```

```
BF2 <- brune.func(spectra$freqs2plot[-1], BF$omega, BF$tstar0, BF$fc, BF$alpha,
BF$gamma)
```

```
plot(BF2, type="l", log="y")
```

```
#Magnitude calculation
```

```
rho= 1190 #m^3/kg
```

```
Vp = 2750 #m/s
```

```
omega = BF$omega0
```

$R_p = 0.52$ #Boore and Boatwright, 1984

$R_s = 0.6$ #Boore and Boatwright, 1984

#Station used is 1/3

station_used <- ceiling (i/3)

#source receiver distance m

$r <- \sqrt{(\text{Locations}\$x[e] - \text{stn_X}[\text{station_used}])^2 + (\text{Locations}\$y[e] - \text{stn_Y}[\text{station_used}])^2 + (\text{Locations}\$z[e] - \text{stn_Z}[\text{station_used}])^2}$

$r_upper <- \sqrt{((\text{Locations}\$x[e] + \text{Loc_error}\$x[e]) - (\text{stn_X}[\text{station_used}] + 0.0005))^2 + ((\text{Locations}\$y[e] + \text{Loc_error}\$y[e]) - (\text{stn_Y}[\text{station_used}] + 0.0005))^2 + ((\text{Locations}\$z[e] + \text{Loc_error}\$z[e]) - (\text{stn_Z}[\text{station_used}] + 0.0005))^2}$

$r_lower <- \sqrt{((\text{Locations}\$x[e] - \text{Loc_error}\$x[e]) - (\text{stn_X}[\text{station_used}] - 0.0005))^2 + ((\text{Locations}\$y[e] - \text{Loc_error}\$y[e]) - (\text{stn_Y}[\text{station_used}] - 0.0005))^2 + ((\text{Locations}\$z[e] - \text{Loc_error}\$z[e]) - (\text{stn_Z}[\text{station_used}] - 0.0005))^2}$

$Mo\$orig[e] = (4 * \pi * \rho * (V_p^3) * r * \omega) / R_p$

$Mo\$upper[e] = (4 * \pi * \rho * (V_p^3) * r_upper * \omega) / R_p$

$Mo\$lower[e] = (4 * \pi * \rho * (V_p^3) * r_lower * \omega) / R_p$

```
#Eaton
```

```
#MwE = (2/3)*log10(Mo)-10.7
```

```
#Baig and Urbancic
```

```
Mw$orig[e] = (2/3)*log10(unlist(Mo$orig[e]))-6
```

```
Mw$upper[e] = (2/3)*log10(unlist(Mo$upper[e]))-6
```

```
Mw$lower[e] = (2/3)*log10(unlist(Mo$lower[e]))-6
```

```
}
```

```
}
```

```
Magnitude <- data.frame(unlist(Mo$orig), unlist(Mo$upper),  
unlist(Mo$lower),unlist(Mw$orig), unlist(Mw$upper),unlist (Mw$lower))
```

```
colnames(Magnitude) <- c("Mo_orig","Mo_upper","Mo_lower",  
"Mw_orig","Mw_upper","Mw_lower")
```

```
write.table(Magnitude,
```

```
file=paste(DIR$DATA_SEISMIC_TXT,
```

```
"/Magnitude.txt",sep=""),  
col.names=TRUE, row.names=TRUE)
```

7. Location steps- PCA_AM.R

The following programs calculate the x,y and z coordinates of the locations of the seismic events and should be used in the order they are listed.

1. Principal Component Analysis (PCA) gives us the azimuth and dip angles that we use in the back projection to find the location of the event. The following script uses PCA and saves the angles for further analysis.

PCA_AM.R:

```
#Principal component analysis
```

```
#Abigail Maxwell
```

```
#October 2015
```

```
setwd("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/1")
```

```
# establish environmental variables
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
data_info <- list()
```

```
decimated_stn <- list()
```

```
filtered_comp2plot <- list()
```

```
opt <- list()
```

```
opt$filter <- FALSE
```

```
azim_pal_axis_deg <- list()
```

```
dip_pal_axis_deg <- list()
```

```
sd_azim_pal_axis_deg <- list()
```

```
sd_dip_pal_axis_deg <- list()
```

```
avg_azim_pal_axis_deg <- list()
```

```
avg_dip_pal_axis_deg <- list()
```

```
azims <- list()
```

```
dips <- list()
```

```
opt$save_events <- F
```

```
opt$plot_PCAsresultsonly <- T
```

```
data_info$sample_rate = 1e06
```

```
data_info$sample_int = 1/data_info$sample_rate
```

```
Eventlist <- read.delim(paste(DIR$DATA_SEISMIC_TXT,"/Events_starttimes_April16.txt",  
sep=""))
```

```
event = seq(from=1, to= 111, by=1)
```

```
for(e in 1:1){
```

```
print(paste("Event ",e, sep=""))
```

```
#Event information
```

```
data_info$t_s_start      = Eventlist$File_Start[[e]]
```

```
data_info$t_s_end        = Eventlist$File_End[[e]]
```

```
data_info$event           = Eventlist$Event[[e]]
```

```
num_samples = (data_info$t_s_end-data_info$t_s_start)*data_info$sample_int +1
```

```
#Bin
```

```
order                = Eventlist$Bin[[e]]
```

```
setwd(paste("/home/abbym/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/1209  
14/",order, sep=""))
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
station_nums <- list()
```

```
station_nums[1]    = "A"
```

```
station_nums[2]    = "B"
```

```
station_nums[3]    = "C"
```

```
station_nums[4]    = "D"
```

```
station_nums[5]    = "E"
```

```
station_nums[6]    = "F"
```

```
station_nums[7]    = "G"
```

```
station_nums[8]    = "H"
```

```
#Read in the data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnA",data_info)
```



```
stnA <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnB",data_info)
```

```
stnB <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnC",data_info)
```

```
stnC <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnD",data_info)
```

```
stnD <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnE",data_info)
```

```
stnE <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnF",data_info)
```

```
stnF <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnG",data_info)
```

```
stnG <- data
```

```
list() -> data
```

```
load_station_events(file.path(DIR$DATA_SEISMIC_RSEIS),base_name="stnH",data_info)
```

```
stnH <- data
```

```
list() -> data
```

#From Xample Galperin Rotate files are saved such that:

#stnA\$Y is actually Z comp

#stnA\$X is actually Y comp

#stnA\$Z is actually X comp

```
STN <- list()
```

```
STN[[1]] <- list()
```

```
STN [[2]]<- list()
```

```
STN[[3]] <- list()
```

```
STN[[4]] <- list()
```

```
STN[[5]] <- list()
```

```
STN[[6]] <- list()
```

```
STN[[7]] <- list()
```

```
STN[[8]] <- list()
```

```
STN[[1]]$Z <- stnA$Y
```

```
STN[[1]]$Y <- stnA$X
```

```
STN[[1]]$X <- stnA$Z
```

```
STN[[2]]$Z <- stnB$Y
```

```
STN[[2]]$Y <- stnB$X
```

```
STN[[2]]$X <- stnB$Z
```

```
STN[[3]]$Z <- stnC$Y
```

```
STN[[3]]$Y <- stnC$X
```

```
STN[[3]]$X <- stnC$Z
```

STN[[4]]\$Z <- stnD\$Y

STN[[4]]\$Y <- stnD\$X

STN[[4]]\$X <- stnD\$Z

STN[[5]]\$Z <- stnE\$Y

STN[[5]]\$Y <- stnE\$X

STN[[5]]\$X <- stnE\$Z

STN[[6]]\$Z <- stnF\$Y

STN[[6]]\$Y <- stnF\$X

STN[[6]]\$X <- stnF\$Z

STN[[7]]\$Z <- stnG\$Y

STN[[7]]\$Y <- stnG\$X

STN[[7]]\$X <- stnG\$Z

STN[[8]]\$Z <- stnH\$Y

STN[[8]]\$Y <- stnH\$X

```
STN[[8]]$X <- stnH$Z
```

```
for (i in 1:1){
```

```
#Plotting the data
```

```
demean_Z <- demean(STN[[i]]$Z)
```

```
demean_Y <- demean(STN[[i]]$Y)
```

```
demean_X <- demean(STN[[i]]$X)
```

```
par(mfrow=c(3,1))
```

```
decimated_stn[[i]] <- list()
```

```
#Decimate the data to make it easier to plot
```

```
decimation_inc = 1000
```

```

decimated_stn[[i]]$X <- decimate_time_series(demean_X,

      from = data_info$t_s_start,

      to   = data_info$t_s_end,

      by   = decimation_inc,

      with = data_info$sample_int)

```

```

decimated_stn[[i]]$Y <- decimate_time_series(demean_Y,

      from = data_info$t_s_start,

      to   = data_info$t_s_end,

      by   = decimation_inc,

      with = data_info$sample_int)

```

```

decimated_stn[[i]]$Z <- decimate_time_series(demean_Z,

      from = data_info$t_s_start,

      to   = data_info$t_s_end,

      by   = decimation_inc,

      with = data_info$sample_int)

```

```

decimated_t      <- decimate_time_series(NULL,

```

```

from = data_info$t_s_start,

to  = data_info$t_s_end,

by  = decimation_inc,

with = data_info$sample_int)

```

```

#Plot Z comp

```

```

plot(decimated_t,decimated_stn[[i]]$Z, type="l",
main=paste("STN",station_nums[i],"",sep=""),

#ylim = PLOT$ylim,

#xlim = PLOT$xlim,

col="red",

xlab = paste("Z- Up ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"

)

```

```

#Plot Y comp

```

```

plot(decimated_t,decimated_stn[[i]]$Y, type="l",
main=paste("STN",station_nums[i],"",sep=""),

#ylim = PLOT$ylim,

```

```

#xlim = PLOT$xlim,

col="blue",

xlab = paste("Y- N ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"

)

```

```

#Plot X comp

```

```

plot(decimated_t,decimated_stn[[i]]$X, type="l",
main=paste("STN",station_nums[i],"",sep=""),

```

```

#ylim = PLOT$ylim,

#xlim = PLOT$xlim,

col="green",

xlab = paste("X- E ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"

)

```

```

if (opt$filter){

```

```

plot.new()

```

```

print("Filtering Data")

```



```
type="BP"
```

```
fl=500
```

```
fh= 5000
```

```
filtered_comp2plot[[i]]<- list()
```

```
filtered_comp2plot[[i]]$X <- butfilt(as.vector(decimated_stn[[i]]$X),fl=fl, fh=fh,  
deltat=data_info$sample_rate, type=type, proto="BU")
```

```
filtered_comp2plot[[i]]$Y <- butfilt(as.vector(decimated_stn[[i]]$Y),fl=fl, fh=fh,  
deltat=data_info$sample_rate, type=type, proto="BU")
```

```
filtered_comp2plot[[i]]$Z <- butfilt(as.vector(decimated_stn[[i]]$Z),fl=fl, fh=fh,  
deltat=data_info$sample_rate, type=type, proto="BU")
```

```
par(mfrow=c(3,1))
```

```
plot(decimated_t,filtered_comp2plot[[i]]$Z , type="l",  
main=paste("STN",station_nums[i],"",sep=""),
```

```
#ylim = PLOT$ylim,
```

```
#xlim = PLOT$xlim,
```

```
col="red",
```

```
xlab = paste("Z- Up ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"
```

)

#Plot Y comp

```
plot(decimated_t,filtered_comp2plot[[i]]$Y, type="l",
```

```
main=paste("STN",station_nums[i],"",sep=""),
```

```
  #ylim = PLOT$ylim,
```

```
  #xlim = PLOT$xlim,
```

```
  col="blue",
```

```
  xlab = paste("Y- N ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"
```

)

#Plot X comp

```
plot(decimated_t,filtered_comp2plot[[i]]$X, type="l",
```

```
main=paste("STN",station_nums[i],"",sep=""),
```

```
  #ylim = PLOT$ylim,
```

```
  #xlim = PLOT$xlim,
```

```
  col="green",
```

```
  xlab = paste("X- E ",data_info$event," Time(s)",sep=" "), ylab="AMP(counts)"
```

)

```
EV <- list()
```

```
PE <- list()
```

```
DT = 1e-3
```

```
NFFT= 500
```

```
Ns = 25
```

```
Nov = 20
```

```
fl = 1
```

```
fh = 500
```

```
EV <- evolfft(filtered_comp2plot[[i]]$X,dt=DT, Nfft=Nfft, Ns=Ns, Nov=Nov, adjust=TRUE,  
fl=fl, fh=fh)
```

```
PE <- plotevol(EV, log=0, fl=1, fh=1000, col=rainbow(100), ygrid=FALSE, STYLE="fft",  
STAMP="")
```

```
#Triming the data
```

```
##Calculating first break value using STA/LTA
```

```
roll_stalta(((filtered_comp2plot[[i]]$X)^2),1,2, increment=1) -> p
```

```
first_break_approx_rel_s = which.max(p)
```

```
#BINFILE
```

```
stn = station_nums[i]
```

```
su_data <- c(filtered_comp2plot[[i]]$Z,filtered_comp2plot[[i]]$Y,filtered_comp2plot[[i]]$X)
```

```
binfile = paste("order",order,"_event",data_info$event,"_STN", stn,".bin",sep="")
```

```
outbound_bin = file.path(DIR$DATA_SEISMIC_SU_PCA,binfile)
```

```
outbound_connection = file(outbound_bin,"wb")
```

```
# output data to su binary file
```

```
writeBin(su_data,outbound_connection, endian="little", size=4)
```

```
close(outbound_connection)
```

```
print(paste("Wrote ",binfile,sep=""))
```

```
#SUPOLAR FUNCTION
```

```
su_num_samples = length(filtered_comp2plot[[i]]$X)
```

```
}
```

```
#if (opt$filter == F)
```

```
else{
```

```
##Calculating first break value using STA/LTA
```

```
roll_stalta((((decimated_stn[[i]]$X)^2),1,2, increment=1) -> p
```

```
first_break_approx_rel_s = which.max(p)
```

```
#BINFILE
```

```
stn = station_nums[i]
```

```
su_data <- c(decimated_stn[[i]]$Z,decimated_stn[[i]]$Y,decimated_stn[[i]]$X)
```

```
binfile      = paste("order",order,"_event",data_info$event,"_STN", stn,".bin",sep="")
```

```
outbound_bin  = file.path(DIR$DATA_SEISMIC_SU_PCA,binfile)
```

```
outbound_connection = file(outbound_bin,"wb")
```

```
# output data to su binary file
```

```
writeBin(su_data,outbound_connection, endian="little", size=4)
```

```
close(outbound_connection)
```

```
print(paste("Wrote ",binfile,sep=""))
```

```
#SUPOLAR FUNCTION
```

```
su_num_samples = length(decimated_stn[[i]]$X)
```

```
}
```

```
binfile      = paste("order",order,"_event",data_info$event,"_STN", stn,sep="")
```

```
setwd(paste(DIR$DATA_SEISMIC_SU_PCA, sep=""))
```

```
system(paste("suaddhead ns=",su_num_samples,"<", binfile,".bin", "| sushw key=dt a=1000  
>",binfile,".su",sep=""))
```

```
system(paste(paste("supolar <", binfile,".su",sep=""), paste("theta=3 phi=3 rl=3 rlq=0.5  
amp=1 angle=deg wl=0.003 win=boxcar verbose=1", sep=" ")))
```

```
#plotting the results of supolar
```

```
#Reading Energy data
```

```
sufile      = "polar.qr"
```

```
inbound     = paste(DIR$DATA_SEISMIC_SU_PCA,  
                    sufile,sep="/")
```

```
connection  <- file(inbound,"rb")
```

```
energy = readSu (connection,su_num_samples)
```

```
close(connection)
```

```
#Reading Azimuth data
```

```
sufile      = "polar.phi"
```

```
inbound      = paste(DIR$DATA_SEISMIC_SU_PCA,  
                     sufile,sep="/")
```

```
connection    <- file(inbound,"rb")
```

```
azimuth= readSu (connection,su_num_samples)
```

```
close(connection)
```

```
#Reading Dip data
```

```
sufile      = "polar.theta"
```

```
inbound      = paste(DIR$DATA_SEISMIC_SU_PCA,  
                     sufile,sep="/")
```

```
connection    <- file(inbound,"rb")
```

```
dip = readSu (connection,su_num_samples)
```

```
close(connection)
```

```
#Finding the peak of Energy
```



```
range_samples = 50
```

```
pos_first_arrival = first_break_approx_rel_s
```

```
early_pos      = pos_first_arrival - range_samples/2
```

```
late_pos       = pos_first_arrival + range_samples/2
```

```
pos_max_Energy_rel = which.max(energy$data[[1]])
```

```
#pos_max_Energy_abs = early_pos + as.numeric(pos_max_Energy_rel) - 1
```

```
pos_max_Energy_abs = which.max(energy$data[[1]])
```

```
#Estimate of azimuth from station to event
```

```
azim_pal_axis_deg[[i]] = azimuth$data[[1]][pos_max_Energy_abs]
```

```
#Average of azimuth
```

```
avg_azim_pal_axis_deg[[i]] = mean(azimuth$data[[1]])
```

```
#Estimate of dip from station to event
```

```
dip_pal_axis_deg[[i]] = dip$data[[1]][pos_max_Energy_abs]
```

```
#Average of dip
```

```
avg_dip_pal_axis_deg[[i]] = mean(dip$data[[1]])
```

```
#plotting the supolar results
```

```
plot.new()
```

```
par(mfrow=c(3,1))
```

```
par(mfg=c(1,1))
```

```
plot(energy$data[[1]],type="l",main="Average energy", ylab="Energy", xlab="Sample  
number")
```

```
points(pos_max_Energy_abs,
```

```
energy$data[[1]][pos_max_Energy_abs],type="p",col="red")
```

```
plot_control(TRUE)
```

```
par(mfg=c(2,1))
```

```
plot(azimuth$data[[1]],type="l",main="Azimuth",ylab="Azimuth (degrees)", xlab="Sample  
number")
```

```

points(pos_max_Energy_abs,

       azim_pal_axis_deg[[i]],type="p",col="red")


plot_control(TRUE)

par(mfg=c(3,1))

plot(dip$data[[1]],ylim=c(0,180),type="l",main="Dip",ylab="Dip (degrees)", xlab="Sample
number")

points(pos_max_Energy_abs,

       dip_pal_axis_deg[[i]], type="p",col="red")


ans = readline_control("Done? [N], or y ")

if(substr(ans,1,1) == "y") {

  questioning = FALSE

  azims[[i]] <- azim_pal_axis_deg[[i]]

  dips[[i]] <- dip_pal_axis_deg[[i]]


  # save output information

  if (opt$save_events) {

```

```
# Write out the back-azimuths to a file and to the terminal
```

```
list() -> event
```

```
# sample window within expected location
```

```
event$range_samples      = range_samples
```

```
# possible location of first break
```

```
event$first_break_approx_rel_s = first_break_approx_rel_s
```

```
# principal axis azimuth in degrees
```

```
event$azim_pal_axis_deg = azim_pal_axis_deg[[i]]
```

```
#principal axis dip in degrees
```

```
event$dip_pal_axis_deg = dip_pal_axis_deg[[i]]
```

```
#Standard deviation of azimuth
```

```
event$sd_azim_pal_axis_deg = sd_azim_pal_axis_deg[[i]]
```

```
#Standard deviation of dip
```

```
event$sd_dip_pal_axis_deg = sd_dip_pal_axis_deg[[i]]
```

```
#Average of azimuth
```

```
event$avg_azim_pal_axis_deg[[i]] = mean(azimuth$data[[1]])
```

```
#Average of dip
```

```
event$avg_dip_pal_axis_deg[[i]] = mean(dip$data[[1]])
```

```
event$station_num = stn
```

```
event$time_span_s = data_info$t_s_end - data_info$t_s_start
```

```
event_file_out = paste("Geo_", "Stn", stn,
```

```
                        "_event_", data_info$event, "_Azim_Dip_PAxis_deg", ".txt", sep="" )
```

```
event_outbound = paste(DIR$DATA_WELL, "/",
```

```
                        "Geo_", "Stn", stn,
```

```
                        "_event_", data_info$event, "_Azim_Dip_PAxis_deg", ".txt", sep="" )
```

```
print(paste("Wrote out ", sep="", event_file_out ))
```

```
write.table(event, file=event_outbound, sep=" ", col.names=TRUE, row.names=FALSE)
```

```

#write.table(event, file="", sep="\t",col.names=TRUE)

}

} else {

    questioning = TRUE

}

#Zoom into energy window

while (questioning) {

    ans = readline("Is this window correct? ")

    if (ans== "y") {

        azims[[i]] <- azim_pal_axis_deg[[i]]

        dips[[i]] <- dip_pal_axis_deg[[i]]

```

```

}

else{

#window_start <- as.numeric(readline ("Enter window start: "))

print("Choose New Window")

window_start <- (readline ("Enter window start: "))

if (window_start == paste("")){

    window_start <- 1

}

else{

    window_start <- as.numeric(window_start)

}

window_end <- (readline ("Enter window end: "))

if (window_end == paste("")){

    window_end <- length(energy$data[[1]])

}

else{

```

```

window_end <- as.numeric(window_end)

}

index <- seq(from= window_start, to= window_end, length=
length(energy$data[[1]][window_start:window_end]))

new_pos_max_Energy_rel <- which.max(energy$data[[1]][window_start:window_end])

new_pos_max_Energy_abs <- new_pos_max_Energy_rel + window_start-1

azims[[i]] <- azimuth$data[[1]][new_pos_max_Energy_abs]

dips[[i]] <- dip$data[[1]][new_pos_max_Energy_abs]

#Standard deviation of azimuth

sd_azim_pal_axis_deg[[i]] <- sd(azimuth$data[[1]][window_start:window_end])

#Standard deviation of dip

sd_dip_pal_axis_deg[[i]] <- sd(dip$data[[1]][window_start:window_end])

#Average of azimuth

```



```
avg_azim_pal_axis_deg[[i]] <- mean(azimuth$data[[1]][window_start:window_end])
```

```
#Average of dip
```

```
avg_dip_pal_axis_deg[[i]] <- mean(dip$data[[1]][window_start:window_end])
```

```
plot.new()
```

```
par(mfrow=c(3,1))
```

```
par(mfg=c(1,1))
```

```
plot(index,energy$data[[1]][window_start:window_end],type="l",main="Average energy",  
ylab="Energy", xlab="Sample number")
```

```
points(new_pos_max_Energy_abs,
```

```
energy$data[[1]][new_pos_max_Energy_abs],type="p",col="red")
```

```
plot_control(TRUE)
```

```
par(mfg=c(2,1))
```

```
plot(index,
```

```
azimuth$data[[1]][window_start:window_end],type="l",main="Azimuth",ylab="Azimuth  
(degrees)", xlab="Sample number")
```

```
points(new_pos_max_Energy_abs,
```

```
      azims[[i]],type="p",col="red")
```

```
plot_control(TRUE)
```

```
par(mfg=c(3,1))
```

```
plot(index,
```

```
dip$data[[1]][window_start:window_end],ylim=c(0,180),type="l",main="Dip",ylab="Dip  
(degrees)", xlab="Sample number")
```

```
points(new_pos_max_Energy_abs,
```

```
      dips[[i]], type="p",col="red")
```

```
}
```

```
ans = readline_control("Done? [N], or y ")
```

```
if(substr(ans,1,1) == "y") {
```

```
  questioning = FALSE
```

```
if (opt$plot_PCAsresultsonly==T) {
```

```

#time <- seq(from=decimated_t>window_start],to=
decimated_t>window_end],length=length(window_start>window_end)) par(mfrow=c(1,1))

par(mfrow=c(1,1))

par(mar=c(5,4,4,8)+0.1)

plot(decimated_t>window_start>window_end],energy$data[[1]][>window_start>window_end
], axes=F,type="l",main="Average energy", ylab="",xlab="Time (s)")

points(new_pos_max_Energy_abs,

       energy$data[[1]][new_pos_max_Energy_abs],type="p",col="red")

axis(1)

axis(2)

mtext("Energy", side=2, line=3)

par(new=T)

plot(decimated_t>window_start>window_end],

azimuth$data[[1]][>window_start>window_end],axes=F,col="blue",type="l",main="",ylab="",
xlab="")

points(new_pos_max_Energy_abs,

       azims[[i]],type="p",col="red")

axis(4)

```

```

mtext("Azimuth (degrees)",side=4,line=2.5)

par(new=T)

plot(decimated_t>window_start>window_end],
dip$data[[1]][window_start>window_end],axes=F,col="green",
ylim=c(0,180),type="l",main="",ylab="", xlab="")

points(new_pos_max_Energy_abs,

      dips[[i]], type="p",col="red")

axis(4, line=4.5)

mtext("Dip (degrees)",side=4,line=6.5)

box(which="plot")

}

if (opt$save_events) {

  # Write out the back-azimuths to a file and to the terminal

  list() -> event

  # sample window within expected location

  event$range_samples      = range_samples

```

```
# possible location of first break
```

```
event$first_break_approx_rel_s = first_break_approx_rel_s
```

```
# principal axis azimuth in degrees
```

```
event$azim_pal_axis_deg = azim_pal_axis_deg[[i]]
```

```
#Standard deviation of azimuth
```

```
event$sd_azim_pal_axis_deg = sd_azim_pal_axis_deg[[i]]
```

```
#Standard deviation of dip
```

```
event$sd_dip_pal_axis_deg = sd_dip_pal_axis_deg[[i]]
```

```
#Average of azimuth
```

```
event$avg_azim_pal_axis_deg = avg_azim_pal_axis_deg[[i]]
```

```
#Average of dip
```

```
event$avg_dip_pal_axis_deg = avg_dip_pal_axis_deg[[i]]
```

```

#principal axis dip in degrees

event$dip_pal_axis_deg = dip_pal_axis_deg[[i]]

event$station_num      = stn

event$time_span_s      = data_info$t_s_end- data_info$t_s_start

event_file_out         = paste("Geo_", "Stn", stn,

                                "_event_", data_info$event, "_Azim_Dip_PAxis_deg", ".txt", sep="" )

event_outbound         = paste(DIR$DATA_WELL, "/",

                                "Geo_", "Stn", stn,

                                "_event_", data_info$event, "_Azim_Dip_PAxis_deg", ".txt", sep="" )

print(paste("Wrote out ", sep="", event_file_out ))

write.table(event, file=event_outbound, sep=" ", col.names=TRUE, row.names=FALSE)

#write.table(event, file="", sep="\t", col.names=TRUE)

}

} else {

```

```
questioning = TRUE
```

```
}
```

8. Location steps: Plot_PCA_pract_120914.R

The following script uses the angles calculated in PCA_AM.R and SVD R package to find the x, y and z coordinates of the location of the event.

```
Plot_PCA_pract_120914.R:
```

```
#Back Propagation using the angles generated from PCA_AM.R
```

```
#Using the SVD method (Han et al., 2010)
```

```
#Abigail Maxwell
```

```
#December 2015
```

```
setwd("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/1")
```

```
# establish environmental variables
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
Eventlist <- read.delim(paste(DIR$DATA_SEISMIC_TXT, "/Events_starttimes_April16.txt",  
sep=""))
```

```
#EVENT <- read.delim(paste(DIR$DATA_SEISMIC_TXT, "/Event_clusters.txt", sep=""))
```

```
order = 1
```

```
EVENT <- seq(from=1, to=111, by=1)
```

```
#EVENT <- EVENT$Event
```

```
#EVENT <- c(2)
```

```
ALL_EVENTS <- list()
```

```
event <- list()
```

```
for (e in 1:1) {
```

```
  #Bin
```

```
  order      = Eventlist$Bin[[e]]
```



```
setwd(paste("/home/abbym/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/1  
20914/",order, sep=""))
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
#event$num = EVENT[e]
```

```
event$num = Eventlist$Event[[e]]
```

```
c  <- list()
```

```
d  <- list()
```

```
stn <- list()
```

```
stn$names <- list("StnA","StnB","StnC","StnD","StnE","StnF","StnG","StnH")
```

```
STN <- list()
```

```
stn$x <- list()
```

```
stn$y <- list()
```

```
stn$z <- list()
```

```
stn$col <- c(1,2,3,4,5,6,7,8)
```

```
stn$AZIM <- list()
```

```
stn$DIP <- list()
```

```
stn$org_AZIM <- list()
```

```
stn$org_DIP <- list()
```

```
stn$L_AZIM <- list()
```

```
stn$U_AZIM <- list()
```

```
stn$L_DIP <- list()
```

```
stn$U_DIP <- list()
```

```
stn$avg_AZIM <- list()
```

```
stn$avg_DIP <- list()
```

```
dir_vect <- list()
```

```
d_points <- list()
```

```
ALL_Poss_Loc <- list()
```

```
ALL_Poss_Loc$x<- list()
```

```
ALL_Poss_Loc$y<- list()
```

```
ALL_Poss_Loc$z<- list()
```

```
#Options for different functions
```

```
opt <- list()
```

```
opt$PLOT = F
```

```
opt$PLOT_ALL_LOC = F
```

```
block_vert <- list()
```

```
Combs <- (rep(list(0), 28))
```

```
num <- list()
```

```
AZIM <- list()
```

```
DIP <- list()
```

```
org_AZIM <- list()
```

```
org_DIP <- list()
```

```
L_AZIM <- list()
```

```
U_AZIM <- list()
```

```
L_DIP <- list()
```

```
U_DIP <- list()
```

```
avg_AZIM <- list()
```

```
avg_DIP <- list()
```

```
a <- list()
```

```
b <- list()
```

```
c <- list()
```

```
stn_X <- list()
```

```
stn_Y <- list()
```

```
stn_Z <- list()
```

```
stn_X[[1]] <- list()
```

```
stn_X[[2]] <- list()
```

```
stn_X[[3]] <- list()
```

```
stn_Y[[1]] <- list()
```

```
stn_Y[[2]] <- list()
```

```
stn_Y[[3]] <- list()
```

```
stn_Z[[1]] <- list()
```

```
stn_Z[[2]] <- list()
```

```
stn_Z[[3]] <- list()
```

```
centre <- list()
```

```
ALL_EVENT_LOC <- list()
```

```

geo_loc <-
read.table(paste("/home/abbym/hydraulicfracturing/well/data/HRKE122/AD_card/sens
or/120914",order,"Geo_XYZ_edited.txt",sep="/"),header=FALSE,
col.names=c("X","Y","Z"))

```

#Using the error in measurement (+/- 0.0005m) for different combinations for the locations

```
stn_X[[1]] <- geo_loc$X - 0.0005
```

```
stn_X[[2]] <- geo_loc$X
```

```
stn_X[[3]] <- geo_loc$X + 0.0005
```

```
stn_Y[[1]] <- geo_loc$Y - 0.0005
```

```
stn_Y[[2]] <- geo_loc$Y
```

```
stn_Y[[3]] <- geo_loc$Y + 0.0005
```

```
stn_Z[[1]] <- (geo_loc$Z) - 0.0007
```

```
stn_Z[[2]] <- (geo_loc$Z)
```

```
stn_Z[[3]] <- (geo_loc$Z) + 0.0007
```

```
for (choice in 2:2){
```

```
  for (stn_coord in 1:8){
```

```
    a[stn_coord] <- stn_X[[choice]][stn_coord]
```

```
    b[stn_coord] <- stn_Y[[choice]][stn_coord]
```

```
    c[stn_coord] <- stn_Z[[choice]][stn_coord]
```

```
  }
```

```
d <- seq(from=0.2, to=-0.78, by=-0.01)
```

```
block_vert$x <- c(0, 0.30, 0, 0.3, 0, 0.30, 0, 0.30)
```

```
block_vert$y <- c(0.15,0.15,0.15,0.15,0,0,0,0)
```

```
block_vert$z <- c(-0.077, -0.077,0.000,0.000,-0.077,-0.077,0.000,0.000)
```

```
for (i in 1:8) {
```

```

STN[[i]] <- read.table(paste(DIR$DATA_WELL,"/Geo_",
stn$names[i],"_event_",event$num,"_Azim_Dip_PAxis_deg", ".txt", sep=""),
header=TRUE)

org_AZIM[i] <- STN[[i]]$azim_pal_axis_deg

org_DIP[i] <- STN[[i]]$dip_pal_axis_deg

L_AZIM[i] <- as.numeric(STN[[i]]$azim_pal_axis_deg)-
as.numeric(STN[[i]]$sd_azim_pal_axis_deg)

L_DIP[i] <- as.numeric(STN[[i]]$dip_pal_axis_deg)-
as.numeric(STN[[i]]$sd_dip_pal_axis_deg)

U_AZIM[i] <-
as.numeric(STN[[i]]$azim_pal_axis_deg)+as.numeric(STN[[i]]$sd_azim_pal_axis_deg)

U_DIP[i] <-
as.numeric(STN[[i]]$dip_pal_axis_deg)+as.numeric(STN[[i]]$sd_dip_pal_axis_deg)

avg_AZIM[i] <- STN[[i]]$avg_azim_pal_axis_deg

avg_DIP[i] <- STN[[i]]$avg_dip_pal_axis_deg

stn$org_AZIM[i] <- STN[[i]]$azim_pal_axis_deg*pi/180

stn$org_DIP[i] <- STN[[i]]$dip_pal_axis_deg*pi/180

```

```

stn$L_AZIM[i] <- as.numeric(L_AZIM[i])*pi/180

stn$L_DIP[i] <- as.numeric(L_DIP[i])*pi/180

stn$U_AZIM[i] <- as.numeric(U_AZIM[i])*pi/180

stn$U_DIP[i] <- as.numeric(U_DIP[i])*pi/180

stn$avg_AZIM[i] <- STN[[i]]$avg_azim_pal_axis_deg*pi/180

stn$avg_DIP[i] <- STN[[i]]$avg_dip_pal_axis_deg*pi/180


AZIMS <- cbind(stn$L_AZIM, stn$org_AZIM, stn$U_AZIM,stn$avg_AZIM)

DIPS <- cbind(stn$L_DIP, stn$org_DIP, stn$U_DIP,stn$avg_DIP)

}


num <- c(1,4,5,7,8)


#num <- c(1,2,3,4,5,6,7,8)


if (length(num)== 1){


combs <- expand.grid(c(1,2,3,4))

```



```
}
```

```
if (length(num)== 2){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 3){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 4){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),
```

```
    c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 5){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),
```

```
    c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 6){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),
```

```
    c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 7){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),
```

```
    c(1,2,3,4),c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (length(num)== 8){
```

```
  combs <- expand.grid(c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),
```

```
    c(1,2,3,4),c(1,2,3,4),c(1,2,3,4),c(1,2,3,4))
```

```
}
```

```
if (choice==1){
```

```

for (i in num){

  if (i %in% c(1,6)){

    AZIM <- as.numeric(AZIMS[,choice][[i]])+ pi/2

    DIP <- DIPS[,choice][[i]]

  }

  if (i %in% c(4,7)){

    AZIM <- -as.numeric(AZIMS[,choice][[i]])+ 2*pi

    DIP <- DIPS[,choice][[i]]

  }

  if ( i%in% c(2,3,5,8)){

    AZIM <- as.numeric(AZIMS[,choice][[i]])

    DIP <- DIPS[,choice][[i]]
  }

```

```
}
```

```
for (r in 1:50) {
```

```
  stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]
```

```
  stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]
```

```
  stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]
```

```
}
```

```
xlims= c(0,0.3)
```

```
ylims= c(0,0.15)
```

```
zlims= c(-0.077,0)
```

```
if(opt$PLOT == T){
```

```
  if (i == 1) {
```

```
    #z= -0.077
```

```
    plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims, type="l",
```

```
col=stn$col[1], box=FALSE)
```

```
    #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[1], box=FALSE)
```

```

    aspect3d("iso")

}

else

{

    #z=-0.077

    plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims,type="l",
col=stn$col[i], box=FALSE, axes=FALSE, add=TRUE)

    #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[i], box=FALSE,
axes=FALSE, add=TRUE)

    aspect3d("iso")

}

for (LOC in 1:8){

    points3d(a[LOC], b[LOC], c[LOC], col="brown", size=6, pch="o", add=TRUE)


    points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)

}

```

```
#plotting sample
```

```
sample <- cube3d(color="red", alpha=0.1)
```

```
sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))
```

```
row.names(sample$vb) <- NULL
```

```
shade3d(sample)
```

```
#plotting plane
```

```
#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 
```

```
#normal to plane -> -0.077, 0.044,  $0.124 \cdot 0.077$ 
```

```
xlim= c(0,0.3)
```

```
ylim= c(0,0.15)
```

```
zlim= c(-0.077,0)
```

```
planes3d(-0.077,0,0.044, $0.124 \cdot 0.077$ ,xlim=xlim, ylim=ylim, zlim=zlim,col="brown",  
alpha= 0.6, add=TRUE)
```

```

    }

}

# Finding intersection point

#number of lines used to find intersection or nearest point !!!!!!!

lines = length(num)

ident <- seq(from=3, by=3, length=8)

R <- matrix(nrow= ident[lines], ncol= 3)

R[is.na(R)] <- 0


col1 <- seq(from=1, by=3, to=24)

col2 <- seq(from=2, by=3, to=24)

col3 <- seq(from=3, by=3, to=24)

```

```
R[col1[1:lines], 1] <- 1
```

```
R[col2[1:lines], 2] <- 1
```

```
R[col3[1:lines], 3] <- 1
```

```
# Define which stations are used
```

```
# Default if all 8 stations are used 1:8;
```

```
#if not then c(list of num of stations used)
```

```
#num <- c(1,2,3,4,5,6,7,8)
```

```
#num <- c(1,4,8)
```

```
for (i in num){
```

```
  if (i %in% c(1,6)){
```

```
    AZIM <- as.numeric(AZIMS[,choice][[i]])+ pi/2
```

```
    DIP <- DIPS[,choice][[i]]
```



```
}
```

```
if (i %in% c(4,7)){
```

```
  AZIM <- -as.numeric(AZIMS[,choice][[i]])+ 2*pi
```

```
  DIP <- DIPS[,choice][[i]]
```

```
}
```

```
if ( i%in% c(2,3,5,8)){
```

```
  AZIM <- as.numeric(AZIMS[,choice][[i]])
```

```
  DIP <- DIPS[,choice][[i]]
```

```
}
```

```
for (r in 1:50) {
```

```
  stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]
```

```
  stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]
```

```
  stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]
```

```

stn$x[r] <- as.numeric(stn$x[r])

stn$y[r] <- as.numeric(stn$y[r])

stn$z[r] <- as.numeric(stn$z[r])

}

```

```

EQN <- list()

EQN <- c(stn$x, stn$y, stn$z)

EQN <- matrix(EQN, nrow=50, ncol=3)

EQN <- as.data.frame(EQN)

colnames(EQN) <- c("X", "Y", "Z")


# EQN <- EQN[EQN$X >= 0 & EQN$X <= 0.151,]

# EQN <- EQN[EQN$Y >= 0 & EQN$Y <= 0.149,]

# EQN <- EQN[EQN$Z >= -0.097 & EQN$Z <= 0.001,]


rownames(EQN) <- seq(from= 1, to= length(EQN$X), by=1)

```

```

    dir_vect[[i]] <- c( as.numeric(unlist(EQN$X[length(EQN$X)]))-
as.numeric(min(unlist(EQN$X[1]))),

                    as.numeric(unlist(EQN$Y[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Y[1]))),

                    as.numeric(unlist(EQN$Z[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Z[1]))))

```

```

d_points[[i]] <- list()

```

```

d_points[[i]] <- c(as.numeric(stn_X[[choice]][i]),

                  as.numeric(stn_Y[[choice]][i]),

                  as.numeric(stn_Z[[choice]][i]))

}

```

```

d_points <- as.matrix(unlist(d_points))

```

```

row <- list()

```

```

column <- seq(from=1, by= 1, length= lines)

```

```
g <- matrix(nrow= ident[lines], ncol= lines)
```

```
g[is.na(g)] <- 0
```

```
for (i in 1:length(num)) {
```

```
  row[[i]] <- seq(from=col1[i], by=1, to=col1[i]+2)
```

```
  g[row[[i]][1],column[i]] <- as.numeric(dir_vect[[num[i]]][1] *-1)
```

```
  g[row[[i]][2],column[i]] <- as.numeric(dir_vect[[num[i]]][2] *-1)
```

```
  g[row[[i]][3],column[i]] <- as.numeric(dir_vect[[num[i]]][3] *-1)
```

```
}
```

```
G <- cbind(R,g)
```

```

print("Calculating SVD")

#Singular Value Decomposition of G

s <- svd(G)

D <- diag(s$d)

m <- (s$v)%*% solve(D) %*% t(s$u)%*%d_points

points3d(m[1,1],m[2,1], m[3,1], size=6, col= "blue", add=TRUE)

par3d("windowRect"=c(0,0,600,600))

par3d("FOV"=30)

par3d("observer"= c(0.0000000, 0.0000000, 0.6648181))

filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_event_",e,".png", sep="")

rgl.snapshot(filename, fmt= "png", top= TRUE)

```

```

ALL_Poss_Loc$x <-m[1,1]

```

```

ALL_Poss_Loc$y <-m[2,1]

```

```

ALL_Poss_Loc$z <-m[3,1]

```

```

centre[[choice]] <- list()

centre[[choice]] <- c(ALL_Poss_Loc$x,ALL_Poss_Loc$y,ALL_Poss_Loc$z)

}

if (choice==2){

print (paste("# of Combinations =", 4^(length(num))))

for(combo in 1:4^(length(num))){

  dir_vect    <- list()

  d_points    <- list()

  #print(combs[combo,])

  #for (i in num){

  for (i in num){

```

```

for (position in 1:length(num)){

  if(num[position]== i){

    position -> pos

    Pos= paste("Pos =",pos)

    #print(Pos)

  }

}

AZIM <- as.numeric(AZIMS[i, combs[combo,pos]])

DIP <- as.numeric(DIPS[i, combs[combo,pos]])

if (i %in% c(1,6)){

  AZIM <- AZIM + pi/2

  DIP <- DIP

}

```

```
if (i %in% c(4,7)){
```

```
  AZIM <- AZIM + 2*pi
```

```
  DIP <- DIP
```

```
}
```

```
if ( i%in% c(2,3,5,8)){
```

```
  AZIM <- AZIM
```

```
  DIP <- DIP
```

```
}
```

```
#Plotting lines
```

```
if (opt$PLOT == TRUE) {
```

```
  for (r in 1:50) {
```

```
    stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]
```

```
    stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]
```

```
    stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]
```



```
}
```

```
xlims= c(0,0.3)
```

```
ylims= c(0,0.15)
```

```
zlims= c(-0.077,0)
```

```
if (i == 1) {
```

```
  #z= -0.077
```

```
  plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims, type="l",  
col=stn$col[1], box=FALSE)
```

```
  #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[1], box=FALSE)
```

```
  aspect3d("iso")
```

```
}
```

```
else
```

```
{
```

```
  #z=-0.077
```

```
  plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims,type="l",  
col=stn$col[i], box=FALSE, axes=FALSE, add=TRUE)
```

```

    #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[i], box=FALSE,
axes=FALSE, add=TRUE)

    aspect3d("iso")

}

for (LOC in 1:8){

points3d(a[LOC], b[LOC], c[LOC], col="brown", size=6, pch="o", add=TRUE)


points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black", size=6,
add=TRUE)

}


#plotting sample

sample <- cube3d(color="red", alpha=0.1)

sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))

row.names(sample$vb) <- NULL

```

```
shade3d(sample)
```

```
#plotting plane
```

```
#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 
```

```
#normal to plane ->  $-0.077, 0.044, 0.124 \cdot 0.077$ 
```

```
planes3d(-0.077,0,0.044,0.124*0.077, col="brown", alpha= 0.6, add=TRUE)
```

```
}
```

```
# Finding intersection point
```

```
#number of lines used to find intersection or nearest point !!!!!!!
```

```
lines = length(num)
```

```
ident <- seq(from=3, by=3, length=8)
```

```
R <- matrix(nrow= ident[lines], ncol= 3)
```

```
R[is.na(R)] <- 0
```

```
col1 <- seq(from=1, by=3, to=24)
```

```
col2 <- seq(from=2, by=3, to=24)
```

```
col3 <- seq(from=3, by=3, to=24)
```

```
R[col1[1:lines], 1] <- 1
```

```
R[col2[1:lines], 2] <- 1
```

```
R[col3[1:lines], 3] <- 1
```

```
# Define which stations are used
```

```
# Default if all 8 stations are used 1:8;
```

```
#if not then c(list of num of stations used)
```

```
#num <- c(1,2,3,4,5,6,7,8)
```

```
#num <- c(1,4,8)
```

```

for (r in 1:50) {

  stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]

  stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]

  stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]


  stn$x[r] <- as.numeric(stn$x[r])

  stn$y[r] <- as.numeric(stn$y[r])

  stn$z[r] <- as.numeric(stn$z[r])

}

```

```

EQN <- list()

```

```

EQN <- c(stn$x, stn$y, stn$z)

```

```

EQN <- matrix(EQN, nrow=50, ncol=3)

```

```

EQN <- as.data.frame(EQN)

```

```

colnames(EQN) <- c("X", "Y", "Z")

```

```

#EQN <- EQN[EQN$X >= 0 & EQN$X <= 0.151,]

```

```

#EQN <- EQN[EQN$Y >= 0 & EQN$Y <= 0.149,]

#EQN <- EQN[EQN$Z >= -0.097 & EQN$Z <= 0.001,]


rownames(EQN) <- seq(from= 1, to= length(EQN$X), by=1)


dir_vect[[i]] <- c(as.numeric(unlist(EQN$X[length(EQN$X)]))-
as.numeric(min(unlist(EQN$X[1]))),

                as.numeric(unlist(EQN$Y[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Y[1]))),

                as.numeric(unlist(EQN$Z[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Z[1]))))


d_points[[i]] <- c(as.numeric(stn_X[[choice]][i]),

                as.numeric(stn_Y[[choice]][i]),

                as.numeric(stn_Z[[choice]][i]))

}


d_points <- as.matrix(unlist(d_points))

```

```
row <- list()
```

```
column <- seq(from=1, by= 1, length= lines)
```

```
g <- matrix(nrow= ident[lines], ncol= lines)
```

```
g[is.na(g)] <- 0
```

```
for (i in 1:length(num)) {
```

```
  row[[i]] <- seq(from=col1[i], by=1, to=col1[i]+2)
```

```
  g[row[[i]][1],column[i]] <- as.numeric(dir_vect[[num[i]]][1]) *-1
```

```
  g[row[[i]][2],column[i]] <- as.numeric(dir_vect[[num[i]]][2]) *-1
```

```
  g[row[[i]][3],column[i]] <- as.numeric(dir_vect[[num[i]]][3]) *-1
```

```
}
```

```
print("Calculating SVD")
```

```
G <- cbind(R,g)
```

```
#s <- svd(G, nu=min(nrow(G),ncol(G)), nv=min(nrow(G),ncol(G)))
```

```
s <- svd(G)
```

```
D <- diag(s$d)
```

```
m <- (s$v)%*% solve(D) %*% t(s$u)%*%d_points
```

```
points3d(m[1,1],m[2,1], m[3,1], size=6, col= "blue", add=TRUE)
```

```
#par3d("windowRect"=c(0,0,600,600))
```

```
#par3d("FOV"=30)
```

```
#par3d("observer"= c(0.0000000, 0.0000000, 0.6648181))
```

```
#filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_event_",e,".png", sep="")
```

```
#rgl.snapshot(filename, fmt= "png", top= TRUE)
```

```
ALL_Poss_Loc$x[[combo]] <-m[1,1]
```

```
ALL_Poss_Loc$y[[combo]] <-m[2,1]
```



```
ALL_Poss_Loc$z[[combo]] <-m[3,1]
```

```
}
```

```
open3d()
```

```
#Plotting Possibilities and error ellipsoid
```

```
plot3d(ALL_Poss_Loc_forplot$x,ALL_Poss_Loc_forplot$y,ALL_Poss_Loc_forplot$z,box=T
```

```
RUE,xlab="x",ylab="y",zlab="z", axes=TRUE)
```

```
for (LOC in 1:8){
```

```
  points3d(a[LOC], b[LOC], c[LOC], col="brown", size=6, pch="o", add=TRUE,axes=FALSE,  
  box=FALSE)
```

```
  #points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",  
  size=6, add=TRUE)
```

```
}
```

```
#plotting sample
```

```
sample <- cube3d(color="red", alpha=0.1)
```

```
sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1))
```

```
row.names(sample$vb) <- NULL
```

```
shade3d(sample)
```

```
#plotting plane
```

```
#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 
```

```
#normal to plane -> -0.077, 0.044,  $0.124 \cdot 0.077$ 
```

```
xlim= c(0,0.3)
```

```
ylim= c(0,0.15)
```

```
zlim= c(-0.077,0)
```

```
planes3d(-0.077,0,0.044, $0.124 \cdot 0.077$ ,xlim=xlim, ylim=ylim, zlim=zlim,col="brown",
```

```
alpha= 0.6, add=TRUE)
```

```
#for (LOC in 1:length(ALL_Poss_Loc$x)){
```

```
# points3d(ALL_Poss_Loc$x[LOC], ALL_Poss_Loc$y[LOC], ALL_Poss_Loc$z[LOC], col=
```

```
"red", size=4, add=TRUE)
```

```
#}
```

```
filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_example2.png", sep="")
```

```
rgl.snapshot(filename, fmt= "png", top= TRUE)
```

```
#ALL_Poss_Loc <- as.data.frame(ALL_Poss_Loc)
```

```
#Plotting all locations before filtering
```

```
#plot3d(ALL_Poss_Loc$x,ALL_Poss_Loc$y,ALL_Poss_Loc$z, )
```

```
#filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_event_",e,".png", sep="")
```

```
#rgl.snapshot(filename, fmt= "png", top= TRUE)
```

```
ALL_Poss_Loc_forplot <- ALL_Poss_Loc
```

```
ALL_Poss_Loc <- unique(ALL_Poss_Loc)
```

```
ALL_Poss_loc_X <- as.matrix(ALL_Poss_Loc$x)
```

```
ALL_Poss_loc_Y <- as.matrix(ALL_Poss_Loc$y)
```

```
ALL_Poss_loc_Z <- as.matrix(ALL_Poss_Loc$z)
```

```
ALL_Poss_Loc_mtx <- as.matrix(ALL_Poss_Loc)
```

```
density_x <- density(ALL_Poss_loc_X)
```

```
density_y <- density(ALL_Poss_loc_Y)
```

```
density_z <- density(ALL_Poss_loc_Z)
```

```
plot(density_x$x,density_x$y, type="l", xlab="x coordinates", ylab="Density")
```

```
points(density_x$x[which.max(density_x$y)], density_x$y[which.max(density_x$y)],  
col="red")
```

```
plot(density_y$x,density_y$y, type="l", xlab="y coordinates", ylab="Density")
```

```
points(density_y$x[which.max(density_y$y)], density_y$y[which.max(density_y$y)],  
col="red")
```

```
plot(density_z$x,density_z$y, type="l", xlab="z coordinates", ylab="Density")
```

```
points(density_z$x[which.max(density_z$y)], density_z$y[which.max(density_z$y)],  
col="red")
```

```
#using the top 10% of the density
```

```
num2use_points_x <- 0.10*(length(density_x$x))
```

```
sd_x <- round((num2use_points_x-1)/2)
```

```
num2use_points_y <- 0.10*length(density_y$x)
```

```
sd_y <- round((num2use_points_y-1)/2)
```

```
num2use_points_z <- 0.10*length(density_z$x)
```

```
sd_z <- round((num2use_points_z-1)/2)
```

```
plot(density_x$x,density_x$y, type="l", xlab="x coordinates", ylab="Density")
```

```
points(density_x$x[which.max(density_x$y)], density_x$y[which.max(density_x$y)],  
col="red")
```

```
points(density_x$x[which.max(density_x$y)-sd_x],density_x$y[which.max(density_x$y)-  
sd_x], col="blue")
```

```
points(density_x$x[which.max(density_x$y)+sd_x],density_x$y[which.max(density_x$y)+  
sd_x], col="blue")
```

```
plot(density_y$x,density_y$y, type="l", xlab="y coordinates", ylab="Density")
```

```

points(density_y$x[which.max(density_y$y)], density_y$y[which.max(density_y$y)],
col="red")

points(density_y$x[which.max(density_y$y)-sd_y],density_y$y[which.max(density_y$y)-
sd_y], col="blue")

points(density_y$x[which.max(density_y$y)+sd_y],density_y$y[which.max(density_y$y)
+sd_y], col="blue")


plot(density_z$x,density_z$y, type="l", xlab="z coordinates", ylab="Density")

points(density_z$x[which.max(density_z$y)], density_z$y[which.max(density_z$y)],
col="red")

points(density_z$x[which.max(density_z$y)-sd_z],density_z$y[which.max(density_z$y)-
sd_z], col="blue")

points(density_z$x[which.max(density_z$y)+sd_z],density_z$y[which.max(density_z$y)+
sd_z], col="blue")


#Clean up locations

#X

density_x <- as.data.frame(cbind(density_x$x, density_x$y))

colnames(density_x) <- c("X", "DENS_X")

```

```
density_x_corr <- density_x[((which.max(density_x$DENS_X)-
sd_x):(which.max(density_x$DENS_X)+sd_x)),]
```

```
ALL_Poss_Loc_corr <- ALL_Poss_Loc[ALL_Poss_Loc$x >= density_x_corr$X[1] &
ALL_Poss_Loc$x <= density_x_corr$X[length(density_x_corr$X)],]
```

```
#Y
```

```
density_y <- as.data.frame(cbind(density_y$x, density_y$y))
colnames(density_y) <- c("Y", "DENS_Y")
```

```
density_y_corr <- density_y[((which.max(density_y$DENS_Y)-
sd_y):(which.max(density_y$DENS_Y)+sd_y)),]
```

```
ALL_Poss_Loc_corr <- ALL_Poss_Loc_corr[ALL_Poss_Loc_corr$y >= density_y_corr$Y[1]
&
```

```
ALL_Poss_Loc_corr$y <=
density_y_corr$Y[length(density_y_corr$Y)],]
```

```
#Z
```

```

density_z <- as.data.frame(cbind(density_z$x, density_z$y))

colnames(density_z) <- c("Z", "DENS_Z")


density_z_corr <- density_z[((which.max(density_z$DENS_Z)-
sd_z):(which.max(density_z$DENS_Z)+sd_z)),]


ALL_Poss_Loc_corr <- ALL_Poss_Loc_corr[ALL_Poss_Loc_corr$z >= density_z_corr$Z[1]
&
                                ALL_Poss_Loc_corr$z <=
density_z_corr$Z[length(density_z_corr$Z)],]


#PLOTING ALL Possible locations for a given event and its ellipsoid


if (opt$PLOT_ALL_LOC==TRUE){

open3d()

for (LOC in 1:8){

```



```

points3d(stn_X[[2]][LOC], stn_Y[[2]][LOC], stn_Z[[2]][LOC], col="brown",size=6,
box=TRUE, axes=TRUE)

#points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)

}

#plotting sample

sample <- cube3d(color="red", alpha=0.1)

sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))

row.names(sample$vb) <- NULL

shade3d(sample)


#plotting plane

#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 

#normal to plane ->  $-0.077, 0.044, 0.124 \cdot 0.077$ 

planes3d(-0.077,0,0.044,0.124*0.077, col="brown", alpha= 0.6, add=TRUE)

#for (LOC in 1:length(ALL_Poss_Loc$x)){

```

```
# points3d(ALL_Poss_Loc$x[LOC], ALL_Poss_Loc$y[LOC], ALL_Poss_Loc$z[LOC], col=
"red", size=4, add=TRUE)
```

```
#}
```

```
#Plotting Possibilities and error ellipsoid
```

```
#points3d(ALL_Poss_Loc$x,ALL_Poss_Loc$y,ALL_Poss_Loc$z)
```

```
#filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_example2.png", sep="")
```

```
#rgl.snapshot(filename, fmt= "png", top= TRUE)
```

```
open3d()
```

```
plot3d(ALL_Poss_Loc_corr$x,ALL_Poss_Loc_corr$y,ALL_Poss_Loc_corr$z,type="p",axes=
TRUE, box=FALSE,xlab="",ylab="",zlab="")
```

```
for (LOC in 1:8){
```

```
  points3d(a[LOC], b[LOC], c[LOC], col="brown", size=6, pch="o", axes=TRUE,
```

```
  box=TRUE,xlab="x",ylab="y",zlab="z")
```

```

#points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)

}

#plotting sample

sample <- cube3d(color="red", alpha=0.1)

sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))

row.names(sample$vb) <- NULL

shade3d(sample)


#plotting plane

#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 

#normal to plane ->  $-0.077, 0.044, 0.124 \cdot 0.077$ 

planes3d(-0.077,0,0.044,0.124*0.077, col="brown", alpha= 0.6, add=TRUE)


#filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_example2.png", sep="")

#rgl.snapshot(filename, fmt= "png", top= TRUE)

```

```

ellips <- ellipse3d(cov(ALL_Poss_Loc_corr),
centre=c(mean(ALL_Poss_Loc_corr$x),mean(ALL_Poss_Loc_corr$y),mean(ALL_Poss_Loc
_corr$z)), level=0.95)

plot3d(ellips, alpha=0.4, col="red", add=TRUE)

filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_example3.png", sep="")

#rgl.snapshot(filename, fmt= "png", top= TRUE)

}

```

```

centre[[choice]] <- list()

centre[[choice]] <-
c(mean(ALL_Poss_Loc_corr$x),mean(ALL_Poss_Loc_corr$y),mean(ALL_Poss_Loc_corr$z)
)

print(centre[[choice]])

```

#plotting sample

```

sample <- cube3d(color="red", alpha=0.1)

sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))

```

```
row.names(sample$vb) <- NULL
```

```
shade3d(sample)
```

```
#plotting plane
```

```
#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 
```

```
#normal to plane -> -0.077, 0.044, 0.124*0.077
```

```
planes3d(-0.077,0,0.044,0.124*0.077, col="brown", alpha= 0.6, add=TRUE)
```

```
# par3d("windowRect"=c(1822,412, 2078,668))
```

```
# par3d("FOV"=30)
```

```
# par3d("observer"= c(0.0000000, 0.0000000, 0.6648181))
```

```
#Prep of data for saving
```

```
event$Location_x[[event$num]] <- mean(ALL_Poss_Loc_corr$x)
```

```
event$Location_y[[event$num]] <- mean(ALL_Poss_Loc_corr$y)
```

```
event$Location_z[[event$num]] <- mean(ALL_Poss_Loc_corr$z)
```

```
Ellips_Cov_mtx <- cov(ALL_Poss_Loc_corr)
```

```
#SAVE LOCATIONS OF EVENTS IN A TEXT FILE
```

```
#1. Save the COV matrix needed for plotting error ellipsoid for each event
```

```
write.table(Ellips_Cov_mtx,
```

```
  file=paste(DIR$DATA_SEISMIC_TXT_LOC,
```

```
    "/Ellips_Cov_mtx_event",event$num,".txt",sep=""),
```

```
  col.names=TRUE, row.names=TRUE)
```

```
#data_frame <- cbind(AZIM,DIP)
```

```
##STN_AZ_DIP <-write.table(data_frame, file=
```

```
(paste("~/abbym/hydraulicfracturing/well/data/HRKE122/AD_card/sensor/120914","/",
```

```
order,"STN_AZ_DIP", colnames=c("AZIM","DIP"))
```

```

# #Plotting ALL FINAL LOCATIONS FOR ALL EVENTS w/ or w/o their error ellipsoids

# ALL_EVENT_LOC <-

read.table("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/r/
events/ALL_EVENT_LOC.txt", header=TRUE, quote="\")

#

}

if (choice==3){

  d_points <- list()

  for (i in length(num)){

    if (i %in% c(1,6)){

      AZIM <- as.numeric(AZIMS[,choice][[i]])+ pi/2

      DIP <- DIPS[,choice][[i]]

    }

    if (i %in% c(4,7)){

```

```
AZIM <- -as.numeric(AZIMS[,choice][[i]])+ 2*pi
```

```
DIP <- DIPS[,choice][[i]]
```

```
}
```

```
if ( i%in% c(2,3,5,8)){
```

```
  AZIM <- as.numeric(AZIMS[,choice][[i]])
```

```
  DIP <- DIPS[,choice][[i]]
```

```
}
```

```
for (r in 1:50) {
```

```
  stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]
```

```
  stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]
```

```
  stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]
```

```
}
```

```
xlims= c(0,0.3)
```

```
ylims= c(0,0.15)
```



```

zlims= c(-0.077,0)

if(opt$PLOT == T){

  if (i == 1) {

    #z= -0.077

    plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims, type="l",
col=stn$col[1], box=FALSE)

    #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[1], box=FALSE)

    aspect3d("iso")

  }

  else

  {

    #z=-0.077

    plot3d(stn$x, stn$y, stn$z, xlims=xlims, ylims=ylims, zlims=zlims,type="l",
col=stn$col[i], box=FALSE, axes=FALSE, add=TRUE)

    #plot3d(stn$x, stn$y, z, xlims=xlims, ylims=ylims,col=stn$col[i], box=FALSE,
axes=FALSE, add=TRUE)

    aspect3d("iso")

  }

```

```

for (LOC in 1:8){

  points3d(a[LOC], b[LOC], c[LOC], col="brown", size=6, pch="o", add=TRUE)


  points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)


}


#plotting sample

sample <- cube3d(color="red", alpha=0.1)

sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))

row.names(sample$vb) <- NULL

shade3d(sample)


}

}

```

```
# Finding intersection point
```

```
#number of lines used to find intersection or nearest point !!!!!!!
```

```
lines = length(num)
```

```
ident <- seq(from=3, by=3, length=8)
```

```
R <- matrix(nrow= ident[lines], ncol= 3)
```

```
R[is.na(R)] <- 0
```

```
col1 <- seq(from=1, by=3, to=24)
```

```
col2 <- seq(from=2, by=3, to=24)
```

```
col3 <- seq(from=3, by=3, to=24)
```

```
R[col1[1:lines], 1] <- 1
```

```
R[col2[1:lines], 2] <- 1
```

```
R[col3[1:lines], 3] <- 1
```

```
# Define which stations are used
```

```
# Default if all 8 stations are used 1:8;
```

```
#if not then c(list of num of stations used)
```

```
#num <- c(1,2,3,4,5,6,7,8)
```

```
#num <- c(1,4,8)
```

```
for (i in num){
```

```
  if (i %in% c(1,6)){
```

```
    AZIM <- as.numeric(AZIMS[,choice][[i]])+ pi/2
```

```
    DIP <- DIPS[,choice][[i]]
```

```
  }
```

```
  if (i %in% c(4,7)){
```

```
AZIM <- -as.numeric(AZIMS[,choice][[i]])+ 2*pi
```

```
DIP <- DIPS[,choice][[i]]
```

```
}
```

```
if ( i%in% c(2,3,5,8)){
```

```
  AZIM <- as.numeric(AZIMS[,choice][[i]])
```

```
  DIP <- DIPS[,choice][[i]]
```

```
}
```

```
for (r in 1:50) {
```

```
  stn$x[r]= d[r]*sin(AZIM)*cos((DIP+(pi/2))) + a[[i]]
```

```
  stn$y[r]= d[r]*cos(AZIM)*cos((DIP+(pi/2))) + b[[i]]
```

```
  stn$z[r]= -d[r]*sin((DIP+(pi/2))) + c[[i]]
```

```
  stn$x[r] <- as.numeric(stn$x[r])
```

```
  stn$y[r] <- as.numeric(stn$y[r])
```

```

    stn$z[r] <- as.numeric(stn$z[r])

}

EQN <- list()

EQN <- c(stn$x, stn$y, stn$z)

EQN <- matrix(EQN, nrow=50, ncol=3)

EQN <- as.data.frame(EQN)

colnames(EQN) <- c("X", "Y", "Z")


# EQN <- EQN[EQN$X >= 0 & EQN$X <= 0.151,]

# EQN <- EQN[EQN$Y >= 0 & EQN$Y <= 0.149,]

# EQN <- EQN[EQN$Z >= -0.097 & EQN$Z <= 0.001,]


rownames(EQN) <- seq(from= 1, to= length(EQN$X), by=1)


dir_vect[[i]] <- c( as.numeric(unlist(EQN$X[length(EQN$X)]))-
as.numeric(min(unlist(EQN$X[1]))),

```

```

as.numeric(unlist(EQN$Y[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Y[1])),

as.numeric(unlist(EQN$Z[length(EQN$X)]))-
as.numeric(min(unlist(EQN$Z[1])))

```

```

d_points[[i]] <- list()

```

```

d_points[[i]] <- c(as.numeric(stn_X[[choice]][i]),

```

```

as.numeric(stn_Y[[choice]][i]),

```

```

as.numeric(stn_Z[[choice]][i]))

```

```

}

```

```

d_points <- as.matrix(unlist(d_points))

```

```

row <- list()

```

```

column <- seq(from=1, by= 1, length= lines)

```

```

g <- matrix(nrow= ident[lines], ncol= lines)

```

```

g[is.na(g)] <- 0

```

```
for (i in 1:length(num)) {
```

```
  row[[i]] <- seq(from=col1[i], by=1, to=col1[i]+2)
```

```
  g[row[[i]][1],column[i]] <- as.numeric(dir_vect[[num[i]]][1] *-1)
```

```
  g[row[[i]][2],column[i]] <- as.numeric(dir_vect[[num[i]]][2] *-1)
```

```
  g[row[[i]][3],column[i]] <- as.numeric(dir_vect[[num[i]]][3] *-1)
```

```
}
```

```
G <- cbind(R,g)
```

```
print("Calculating SVD")
```

```
#Singular Value Decomposition of G
```

```
s <- svd(G)
```



```
D <- diag(s$d)
```

```
m <- (s$v)%*% solve(D) %*% t(s$u)%*%d_points
```

```
points3d(m[1,1],m[2,1], m[3,1], size=6, col= "blue", add=TRUE)
```

```
#par3d("windowRect"=c(0,0,600,600))
```

```
#par3d("FOV"=30)
```

```
#par3d("observer"= c(0.0000000, 0.0000000, 0.6648181))
```

```
#filename <- paste(DIR$IMAGES_SEISMIC_JPEG_LOC, "/Loc_event_",e,".png", sep="")
```

```
#rgl.snapshot(filename, fmt= "png", top= TRUE)
```

```
Poss_Loc <- list()
```

```
Poss_Loc$x <-m[1,1]
```

```
Poss_Loc$y <-m[2,1]
```

```
Poss_Loc$z <-m[3,1]
```

```
centre[[choice]] <- list()
```

```
centre[[choice]] <- c(Poss_Loc$x,Poss_Loc$y,Poss_Loc$z)
```

```
}
```

```
}
```

```
#Saving each location- upper, actual, lower limit
```

```
ALL_EVENT_LOC$UP[[event$num]] <- c(centre[[1]])
```

```
ALL_EVENT_LOC$ACTUAL[[event$num]] <- c(centre[[2]])
```

```
ALL_EVENT_LOC$LOW[[event$num]] <- c(centre[[3]])
```

```
#write.table(cbind(event$num,ALL_EVENT_LOC$UP[[event$num]],ALL_EVENT_LOC$ACT  
UAL[[event$num]],ALL_EVENT_LOC$LOW[[event$num]]),
```

```
  # file=paste(DIR$DATA_SEISMIC_TXT_LOC,
```

```
    #"/ALL_EVENT_LOC_event",event$num,".txt",sep=""),
```

```
  #col.names=TRUE, row.names= TRUE)
```

```
#2. Save the location obtained which is the centre of the ellipsoid
```

```
# write.table(cbind(event$num,event$Location_x,event$Location_y,event$Location_z),
```

```

#         file=paste(DIR$DATA_SEISMIC_TXT_LOC,

#             "ALL_EVENT_LOC.txt",sep=""),

#         col.names=TRUE, row.names= TRUE)


# #w/o error ellipsoid

# open3d()

# for (LOC in 1:8){

#   points3d(a[LOC], b[LOC], c[LOC], col="brown",size=6, box=FALSE)

#

#   points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)

#

# }

#

# for (LOC in 1:length(ALL_EVENT_LOC$V1)){

#   points3d(ALL_EVENT_LOC$V2[LOC], ALL_EVENT_LOC$V3[LOC],
ALL_EVENT_LOC$V4[LOC], col= "red", size=4, add=TRUE)

#

```

```

# }

#

# #w/ error ellipsoid

# #read in cov matrix for event

# open3d()

# for (LOC in 1:8){

#   points3d(a[LOC], b[LOC], c[LOC], col="brown",size=6, box=FALSE)

#

#   points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",
size=6, add=TRUE)

#

# }

#

#

# for (e in EVENT){

# Ellips_Cov_mtx<-
read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/11051
4/1/r/events/Ellips_Cov_mtx_",event$num, sep=""), header=TRUE, quote="\")

#

```

```

# for (LOC in 1:length(ALL_EVENT_LOC$V1)){

#   points3d(ALL_EVENT_LOC$V2[LOC], ALL_EVENT_LOC$V3[LOC],
ALL_EVENT_LOC$V4[LOC], col= "red", size=4, add=TRUE)

#   ellips <- ellipse3d(cov(ALL_Poss_Loc_corr),
centre=c(ALL_EVENT_LOC$V2[LOC],ALL_EVENT_LOC$V3[LOC],ALL_EVENT_LOC$V4[LOC]
), level=0.95)

#   plot3d(ellips, alpha=0.8, col="red", add=TRUE)

# }

}

}

setwd(paste("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/",order,
sep=""))

}

}

```

9. Location steps: plotting_locations_AM.R

The following script plots the coordinates on a diagram of the block.

plotting_locations_AM.R :

```
#Plotting locations on a schematic diagram of the sample
```

```
#Abigail Maxwell
```

```
#January 2016
```

```
setwd("~/hydraulicfracturing/seismics/r/HRKE122/AD_card/sensor/120914/1")
```

```
# establish environmental variables
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
block_vert <- list()
```

```
order <- list()
```

```
event <- list()
```

```
max_x <- list()
```

```
max_y <- list()
```

```
max_z <- list()
```

```
x_witherrors <- list()
```

```
y_witherrors <- list()
```

```
z_witherrors <- list()
```

```
x_withouterrors <- list()
```

```
y_withouterrors <- list()
```

```
z_withouterrors <- list()
```

```
opt <- list()
```

```
opt$plotloc_ellipsoid =T
```

```
opt$save_locationswitherror=F
```

```
opt$plot_xycoordinates=F
```

```
Eventlist <- read.delim(paste(DIR$DATA_SEISMIC_TXT,"/Events_starttimes_April16.txt",  
sep=""))
```

```
#EVENT <- read.delim(paste(DIR$DATA_SEISMIC_TXT,"/Event_clusters.txt", sep=""))
```

```
EVENT <-1:111
```

```
if (opt$plotloc_ellipsoid ==T){
```

```
for (e in EVENT[104]){
```

```
  #for (e in EVENT[-c(109,110)]){
```

```

geo_loc <-

read.table(paste("/home/abbym/hydraulicfracturing/well/data/HRKE122/AD_card/sens
or/120914",1,"Geo_XYZ_edited.txt",sep="/"),header=FALSE, col.names=c("X","Y","Z"))

block_vert$x <- c(0, 0.30, 0, 0.3, 0, 0.30, 0, 0.30)

block_vert$y <- c(0.15,0.15,0.15,0.15,0,0,0,0)

block_vert$z <- c(-0.077, -0.077,0.001,0.001,-0.077,-0.077,0.001,0.001)


stn_X <- geo_loc$X

stn_Y <- geo_loc$Y

stn_Z <- geo_loc$Z


zoom <- par3d()$zoom

userMatrix <- par3d()$userMatrix

windowRect <-par3d()$windowRect

```



```
par3d(zoom=zoom, userMatrix= userMatrix, windowRect=windowRect)
```

```
#par3d()
```

```
#windowRect <- c(166,210,527,485)
```

```
#zoom= 0.7
```

```
#userMatrix <- matrix(c(1,0,0,0,0,0.34,0.939,0,0,-0.939,0.34,0,0,0,0,1), nrow=4,  
ncol=4,byrow=TRUE)
```

```
#par3d(windowRect=windowRect, zoom=zoom, userMatrix=userMatrix)
```

```
xlims= c(0,0.3)
```

```
ylims= c(0,0.15)
```

```
zlims= c(-0.077,0)
```

```
#for (LOC in 1:8){
```

```
  plot3d(stn_X, stn_Y, stn_Z,col="brown",size=6, xlab="x", ylab="y", zlab="z",box=TRUE,  
axes=TRUE)
```

```
  #points3d(block_vert$x[LOC], block_vert$y[LOC], block_vert$z[LOC], col="black",  
size=6, add=TRUE)
```

```
#}
```

```
#plotting sample
```

```
sample <- cube3d(color="red", alpha=0.1)
```

```
sample$vb <- rbind(block_vert$x,block_vert$y,block_vert$z, c(1,1,1,1,1,1,1,1))
```

```
row.names(sample$vb) <- NULL
```

```
shade3d(sample)
```

```
#plotting plane
```

```
#eqn of plane  $-0.077x + 0.044z = -0.124 \cdot 0.077$ 
```

```
#normal to plane -> -0.077, 0.044,  $0.124 \cdot 0.077$ 
```

```
planes3d(-0.077,0,0.044, $0.124 \cdot 0.077$ , col="brown", alpha= 0.6, add=TRUE)
```

```
# plotting borehole centres
```

```
borehole_x <- c(0.164,0.164,0.164)
```

```
borehole_y <- c(0.077, 0.077, 0.077)
```

```
borehole_z <- c(0,-0.029, -0.039)
```

```

#plot3d(borehole_x, borehole_y, borehole_z, add=TRUE, lty=2, lwd=4, type="l",
col="blue")

#points3d(borehole_x[1], borehole_y[1], borehole_z[1], add=TRUE, pch="x", cex=20,
col="blue")


#EVENT <-c(10,26,31,34,39,50,57,72,101)

#for (e in 1:10){

    print(e)

    #for (e in EVENT[-c(109,110)]){

        # for (e in EVENT[1]){

            order[e] = Eventlist$Bin[e]

            event[e] = Eventlist$Event[e]

```

```

event_locations <-

read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/ALL_EVENT_LOC_event",event[e],".txt", sep=""),header=TRUE,
quote="\")

#event_locations_upper<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_upperlimit.txt")

#event_locations_lower<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_lower.txt")


Ellips_Cov_mtx_event <-

read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/Ellips_Cov_mtx_event",event[e],".txt", sep=""), header=TRUE,
quote="\")


as.matrix(Ellips_Cov_mtx_event) -> Ellips_Cov_mtx_event

#Plotting Possibilities and error ellipsoid

centre <- c(event_locations$V3[1],event_locations$V3[2],event_locations$V3[3])

ellips <- ellipse3d(Ellips_Cov_mtx_event, centre= centre, level=0.95)

plot3d(ellips, alpha=0.8, col="red", add=TRUE)

```

```
axes <- ellipse3d.axes(Ellips_Cov_mtx_event, centre= centre, level=0.95, colour="gray",
lwd=2)
```

```
points3d(centre[1], centre[2], centre[3])
```

```
# length_xaxis <- sqrt((axes[2,1]-axes[1,2])^2+(axes[4,1]-axes[3,1])^2
```

```
#           +(axes[6,1]-axes[5,1])^2)
```

```
# length_xaxis <- sqrt((axes[2,1]-axes[1,2])^2+(axes[4,1]-axes[3,1])^2
```

```
#           +(axes[6,1]-axes[5,1])^2)
```

```
# length_xaxis <- sqrt((axes[2,1]-axes[1,2])^2+(axes[4,1]-axes[3,1])^2
```

```
#           +(axes[6,1]-axes[5,1])^2)
```

```
x_diff <- c(centre[1]-axes[1,1], centre[1]-axes[2,1], centre[1]-axes[3,1],
```

```
          centre[1]-axes[4,1], centre[1]-axes[5,1], centre[1]-axes[6,1])
```

```
max_x[e] <- round(max(x_diff),4)
```

```
y_diff <- c(centre[2]-axes[2,2], centre[2]-axes[3,2], centre[2]-axes[3,2],
```

```
          centre[2]-axes[4,2], centre[2]-axes[5,2], centre[2]-axes[6,3])
```

```
max_y[e] <- round(max(y_diff),4)
```

```

z_diff <- c(centre[3]-axes[1,3], centre[3]-axes[2,3], centre[3]-axes[3,3],

           centre[3]-axes[4,3], centre[3]-axes[5,3], centre[3]-axes[6,3])

max_z[e] <- round(max(z_diff),4)


#points3d(event_locations$V3[1],event_locations$V3[2],event_locations$V3[3],size=4,
add=TRUE)

#points3d(event_locations_upper$x,event_locations_upper$y,event_locations_upper$z,
size=4,col="red", add=TRUE)

#points3d(event_locations_lower$x,event_locations_lower$y,event_locations_lower$z,
size=4,col="blue", ax

centre[1] <- round(centre[1], 4)

centre[2] <- round(centre[2], 4)

centre[3] <- round(centre[3], 4)


x_withouterrors[e] <- centre[1]

y_withouterrors[e] <- centre[2]

z_withouterrors[e] <- centre[3]

```

```
x_witherrors[e] <- paste(centre[1], "\u00b1", max_x[e])
```

```
y_witherrors[e] <- paste(centre[2], "\u00b1", max_y[e])
```

```
z_witherrors[e] <- paste(centre[3], "\u00b1", max_z[e])
```

```
file_name <-paste(DIR$IMAGES_SEISMIC_JPEG,  
"/HRKE122/AD_card/sensor/120914/1/locations/updatedlocationsEvent_",e,".png",sep  
="")
```

```
#file_name <-paste(DIR$IMAGES_SEISMIC_JPEG,  
"/HRKE122/AD_card/sensor/120914/1/locations/ALLEvents2_errors.png",sep="")
```

```
rgl.snapshot(file_name, fmt="png", top=TRUE)
```

```
#rgl.close()
```

```
}
```

```
}
```

```
if(opt$save_locationswitherror==T){
```

```
x_witherrors <- unlist(x_witherrors)
```

```
y_witherrors <- unlist(y_witherrors)
```

```
z_witherrors <- unlist(z_witherrors)
```

```
Encoding(x_witherrors[e]) <- "UTF-8"
```

```
Encoding(y_witherrors[e]) <- "UTF-8"
```

```
Encoding(z_witherrors[e]) <- "UTF-8"
```

```
locations_withouterrors <- data.frame(unlist(x_withouterrors),  
unlist(y_withouterrors),unlist(z_withouterrors))
```

```
locations_witherrors <- data.frame(x_witherrors, y_witherrors,z_witherrors)
```

```
loc_errors <- data.frame(unlist(max_x),unlist(max_y),unlist(max_z))
```

```
colnames(locations_witherrors) <- c("x","y","z")
```

```
write.table(locations_witherrors,
```

```
file=paste(DIR$DATA_SEISMIC_TXT_LOC,
```

```
"/ALLLocations_errors.txt",sep=""),
```

```
col.names=TRUE, row.names=TRUE)
```



```

colnames(locations_withouterrors) <- c("x","y","z")

write.table(locations_withouterrors,

            file=paste(DIR$DATA_SEISMIC_TXT_LOC,

                        "/ALLLocations_withouterrors.txt",sep=""),

            col.names=TRUE, row.names=TRUE)

colnames(loc_errors) <- c("x_errors","y_errors","z_errors")

write.table(loc_errors,

            file=paste(DIR$DATA_SEISMIC_TXT_LOC,

                        "/Loc_errors.txt",sep=""),

            col.names=TRUE, row.names=TRUE)

}

if(opt$plot_xycoordinates==T){

for (e in 1:111){

order[e] = Eventlist$Bin[e]

event[e] = Eventlist$Event[e]

```

```

event_locations <-
read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/ALL_EVENT_LOC_event",event[e],".txt", sep=""),header=TRUE,
quote="\\"")

centre <- c(event_locations$V3[1],event_locations$V3[2],event_locations$V3[3])

if(e ==1){

plot(event_locations$V3[1], event_locations$V3[2], xlab="Event Location, x coordinate
(m)", ylab="Event Location, y coordinate (m)", type="p")

}

points(centre[1], centre[2])

}

for(e in EVENT[-c(109,110)]){

order[e] = Eventlist$Bin[e]

event[e] = Eventlist$Event[e]

```

```

event_locations <-
read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/ALL_EVENT_LOC_event",event[e],".txt", sep=""),header=TRUE,
quote="\\"")

```

```

centre <- c(event_locations$V3[1],event_locations$V3[2],event_locations$V3[3])

```

```

if(e ==1){

```

```

  plot(Eventlist$Event_Start[1], event_locations$V3[3], xlab="Time (s)", ylab="Event
  Location, y coordinate (m)", type="p", ylim=c(-0.07,-0.03))

}

```

```

  points(Eventlist$Event_Start[e], centre[3])

```

```

}

```

```

}

```

10. The following script plots the x, y and z coordinates of the microseismic event on snapshots from the video recordings. The snapshots reflect the time during the experiment that the specific event plotted occurs.

Plotting_locations_imageAM.R :

```
#Plotting Locations on images
```

```
#Author: Abigail Maxwell
```

```
#Sept 4th 2016
```

```
DIR          <- Project_Dirs("Project_Variables")
```

```
order <- list()
```

```
event <- list()
```

```
opt <- list()
```

```
event_type <- list()
```

```
opt$ploteventtypes=T
```

```
opt$plotlocsthesame=F
```

```
Eventlist <- read.delim(paste(DIR$DATA_SEISMIC_TXT,"/Events_starttimes_April16.txt",  
sep=""))
```

```

EVENT <- read.delim(paste(DIR$DATA_SEISMIC_TXT,"/Event_clusters.txt", sep=""))

EVENT <- EVENT$Event

#Events

Events <-

read.delim(paste("/home/abbym/hydraulicfracturing/seismics/data/HRKE122/AD_card/
sensor/120914/1/txt/Events_starttimes_April16.txt"), header=TRUE)

#LF_cluster events

LF_cluster <-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/Eventlist/LF_cluster_updated.txt")

#LF_nocluster events

LF_nocluster <-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/Eventlist/LF_nocluster_updated.txt")

#HF_cluster events

HF_cluster <-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/Eventlist/HF_cluster_updated.txt")

#HF_nocluster events

```

```

HF_nocluster <-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/Eventlist/HF_nocluster_updated.txt")


event <- 1:111

if (opt$plotlocsthesame==T){

#events_ampover1000 <- c(2,9,10, 23,29,30,31,35,40,54,64,66,72,86,87,88,92)

#events_ampover1000 <-c(10)

for (e in 1:1){

  #png(paste(DIR$IMAGES_SEISMIC_JPEG_CRACK_CRACKPICTS,
"/Picts_withlocs/Event",e,"_pictslocs.png",sep=""), width=3, height=3, units="in",
res=1200)


crack_img <-

readPNG(paste(DIR$IMAGES_SEISMIC_JPEG_CRACKPICTS,"/Event",e,"_edited.png",sep=
""))

plot(0.06:0.186,0.043:0.169, type="n", xlab="x (m)", ylab="y (m)",ylim=c(0.043,0.107),
xlim=c(0.06,0.186), xaxs="i", yaxs="i",xaxt="n", main=paste("Event",e, sep=" "))

axis(1,xaxp=c(0.06,0.18,12))

axis(1, ,at=seq(from=0.06,to=0.186,by=0.01), labels=FALSE, tcl=-0.2)

```

```
axis(1, at=seq(from=0.06,to=0.186,by=0.001),labels=FALSE,tcl=-0.2)
```

```
axis(1, at=seq(from=0.06,to=0.186,by=0.005),labels=FALSE,tcl=-0.3)
```

```
axis(2, at=seq(from=0.043,to=0.107,by=0.001),labels=FALSE,tcl=-0.2)
```

```
axis(2, at=seq(from=0.04,to=0.107,by=0.005),labels=FALSE,tcl=-0.3)
```

```
lim <- par()
```

```
rasterImage(crack_img,lim$usr[1], lim$usr[3], lim$usr[2],lim$usr[4])
```

```
#minor.tick(ny=2, tick.ratio=0.6)
```

```
#minor.tick(nx=4, tick.ratio=0.6)
```

```
#minor.tick(nx=20, ny=10, tick.ratio=0.4)
```

```
order[e] = Eventlist$Bin[e]
```

```
event[e] = Eventlist$Event[e]
```

```

event_locations <-

read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/ALL_EVENT_LOC_event",event[e],".txt", sep=""),header=TRUE,
quote="\")

#event_locations_upper<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_upperlimit.txt")

#event_locations_lower<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_lower.txt")


Ellips_Cov_mtx_event <-

read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/Ellips_Cov_mtx_event",event[e],".txt", sep=""), header=TRUE,
quote="\")


as.matrix(Ellips_Cov_mtx_event[c(1,2), c(1,2)]) -> Ellips_Cov_mtx_event

#Plotting Possibilities and error ellipsoid

centre <- as.vector(c(event_locations$V3[1],event_locations$V3[2]))

#ellips <- ellipse(Ellips_Cov_mtx_event, center= centre, level=0.95,
t=sqrt(qchisq(0.95,2)),)

```



```

ellips <- ellipse(centre, Ellips_Cov_mtx_event,1,add=TRUE,center.pch="x",
center.cex=1,lty=2,fill=TRUE, fill.alpha=0.3)

#par(new=T)

#lines(ellips,col="red", ylim=c(0.043,0.107), xlim=c(0.06,0.186),xaxs="i", yaxs="i",
type="l")

#points(centre[1],centre[2], col="red", pch="x",cex=2)

#axes <- ellipse3d.axes(Ellips_Cov_mtx_event, centre= centre, level=0.95,
colour="gray", lwd=2)

}

}

#Plotting events colour coordinated with symbols from the graphs

if (opt$ploteventtypes==T){

type_names <- c( "LF cluster", "LF non-cluster", "HF cluster", "HF non-cluster")

file_names <- c("LF_cluster", "LF_nocluster","HF_cluster","HF_nocluster")

event_type <- c(LF_cluster, LF_nocluster, HF_cluster, HF_nocluster)

cols <- c("dark green", "dark green", "purple", "purple")

```

```

symbols <- c(8,12,8,12)

for (i in 1:4){

  for (e in unlist(event_type[i])){

    png(paste(DIR$IMAGES_SEISMIC_JPEG_CRACKPICTS,"/Picts_withlocs/Event",e,"_",
file_names[i],"_pictslocs.png",sep=""), width=680, height=500, res=100)

    crack_img <-
readPNG(paste(DIR$IMAGES_SEISMIC_JPEG_CRACKPICTS,"/Event",e,"_edited.png",sep=
""))

    plot(0.06:0.186,0.043:0.169, type="n", xlab="x (m)", ylab="y (m)",ylim=c(0.043,0.107),
xlim=c(0.06,0.186), xaxs="i", yaxs="i",xaxt="n", main=paste("Event ",e,"-
",type_names[i], sep=""))

    axis(1,xaxp=c(0.06,0.18,12))

    axis(1, ,at=seq(from=0.06,to=0.186,by=0.01), labels=FALSE, tcl=-0.2)

    axis(1, at=seq(from=0.06,to=0.186,by=0.001),labels=FALSE,tcl=-0.2)

    axis(1, at=seq(from=0.06,to=0.186,by=0.005),labels=FALSE,tcl=-0.3)

    axis(2, at=seq(from=0.043,to=0.107,by=0.001),labels=FALSE,tcl=-0.2)

```

```
axis(2, at=seq(from=0.04,to=0.107,by=0.005),labels=FALSE,tcl=-0.3)
```

```
lim <- par()
```

```
rasterImage(crack_img,lim$usr[1], lim$usr[3], lim$usr[2],lim$usr[4])
```

```
#minor.tick(ny=2, tick.ratio=0.6)
```

```
#minor.tick(nx=4, tick.ratio=0.6)
```

```
#minor.tick(nx=20, ny=10, tick.ratio=0.4)
```

```
order[e] = Eventlist$Bin[e]
```

```
event[e] = Eventlist$Event[e]
```

```
event_locations <-
```

```
read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091  
4/",order[e],"/txt/loc/ALL_EVENT_LOC_event",event[e],".txt", sep=""),header=TRUE,  
quote="\")
```

```

#event_locations_upper<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_upperlimit.txt")

#event_locations_lower<-

read.delim("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/120914/1/t
xt/loc/120914_locations_lower.txt")


Ellips_Cov_mtx_event <-

read.table(paste("~/hydraulicfracturing/seismics/data/HRKE122/AD_card/sensor/12091
4/",order[e],"/txt/loc/Ellips_Cov_mtx_event",event[e],".txt", sep=""), header=TRUE,
quote="\")

as.matrix(Ellips_Cov_mtx_event[c(1,2), c(1,2)]) -> Ellips_Cov_mtx_event

#Plotting Possibilities and error ellipsoid

centre <- as.vector(c(event_locations$V3[1],event_locations$V3[2]))

#ellips <- ellipse(Ellips_Cov_mtx_event, center= centre, level=0.95,
t=sqrt(qchisq(0.95,2)),)

ellips <- ellipse(centre, Ellips_Cov_mtx_event,1,add=TRUE,center.pch=symbols[i], col=
cols[i],center.cex=1.1,lty=2,lwd=1.5,fill=TRUE, fill.alpha=0.4)

#par(new=T)

```

```

#lines(ellips,col="red", ylim=c(0.043,0.107), xlim=c(0.06,0.186),xaxs="i", yaxs="i",
type="l")

#points(centre[1],centre[2], col="red", pch="x",cex=2)

#

#axes <- ellipse3d.axes(Ellips_Cov_mtx_event, centre= centre, level=0.95,
colour="gray", lwd=2)

dev.off()

}

}

}

```

Vita

Abigail Maxwell, a resident of Trinidad and Tobago, completed her Bachelor of Science in Geology, with a minor in Mathematics, at the University of Georgia, Athens, Georgia, U.S.A., in 2013. She completed a senior thesis on the Composition of the Appalachian Crust from Seismic Data which peaked her interest in pursuing Geophysics for her Masters. Abigail joined the Department of Geology and Geophysics at LSU in August 2014 for her Master's in Geophysics to continue her interests. She is currently employed with BP Trinidad and Tobago as a Challenge Geophysicist.