Acoustic emission detection of micro-cracks under high pressure and high temperature in a deformation large-volume apparatus

Shuailing Ma[1,2,3], Julien Gasc[4], Sarah Incel[5], Stefan Sonntag[1], Robert Farla[1]

1. Deutsches Elektronen-Synchrotron DESY, Notkestr. 85, 22607 Hamburg, Germany

2. Synergetic Extreme Condition High-Pressure Science Center,State Key Laboratory of Superhard Materials, college of physics, Jilin University, Changchun 130012, China

3. Now at: Institute of High Pressure Physics, School of Physical Scientific and Technology, Ningbo University, Ningbo, 315211, China

4. Laboratoire de Géologie, CNRS – École Normale Supérieure, 24 rue Lhomond, 75005 Paris, France

5. Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany

Correspondence email: robert.farla@desy.de

Abstract (250 words) We successfully developed an *in-situ* Acoustic Emission (AE) detection setup that allows recording of AE waveforms (triggered and streaming) and simultaneous X-ray diffraction and imaging on samples deformed at high pressure and high temperature (HPHT) conditions in the Aster-15 LVP at the synchrotron beamline station P61B. This high-pressure AE detection system is a powerful tool to investigate AE phenomena from the HPHT chamber. Six commercial acoustic sensors, protected by a tungsten carbide support ring on each anvil of the same material, have excellent survivability throughout each successive experiment. By pulsing each sensor in succession, the average wave velocity through the anvils and cell assembly can be determined at any press load. The distance between the sensors is obtained by X-ray radiography and by logging the positions of each hydraulic ram. This provides a basis for accurately locating AE events in the sample. The feasibility of this AE detection setup was confirmed by compression and deformation test runs using several different self-designed AE sources in specialized assemblies. The present setup proves to be extremely efficient and accurate in measuring brittle processes in samples under HPHT. It is now available for applications for beam time and experiments without X-rays at P61B. Combined with synchrotron X-rays, *in-situ* pressure, temperature, strain rate and stress, and phase changes can be monitored, while recording AE activity. We provide a powerful tool to investigate the origin of earthquakes, for example causing AE emissions due to brittle dehydration reactions or phase transformations in the Earth.

Keywords: extreme conditions, Acoustic Emissions, high-pressure, rock deformation, energy-dispersive X-ray diffraction, radiography, micro-cracking, earthquakes

1. Introduction

High pressure and high temperature (HPHT) generation is an important means for preparing and studying material properties and simulating Earth's interior environment[1–3]. There are two different categories of high-pressure apparatus, diamond anvil cell (DAC) and large volume press (LVP). Due to high light transmittance of the diamond anvils in the DAC, it can be coupled with a variety of spectroscopy measurement methods, and pressure and temperature generation has been greatly expanded[4]. In the LVP, much larger samples up to mm-size are surrounded by various components, including hard tungsten carbide anvils, pressure transmission media (PTM), insulation sleeves, and a resistive heater. This configuration poses challenges to probe the sample by X-rays. To date, the most common *in situ* synchrotron techniques used in the LVP are X-ray diffraction and imaging. These two methods can be combined using other *in situ* techniques such as ultrasonic interferometry (wave speed measurements)[5], and electrical conductivity measurements[6]. More recently, a third *in situ* technique was developed to explore the internal development in the sample at HPHT: the recording of microscopic fracturing due to volume changes from phase transitions or breakdown of hydrous minerals with and without an applied deviatoric stress[7]. This promising new technique is now also developed at the LVP station P61B at PETRA III, DESY.

The technique is called acoustic emissions (AE) testing, and offers the capability to record and locate the sources of fast-moving cracks in a material. In other words, when a crack propagates, it releases stored strain energy as elastic waves, which are detectable by acoustic sensors (typically a piezoelectric material). AE detection is already widely used in two areas: non-destructive evaluation testing (NDT) and studies used for materials design, and concrete, metallic or composite structure monitoring[8–11]. A new application of AE detection in the laboratory using an LVP not only identifies embrittlement of a ceramic under pressure, temperature and stress, but may be extended to detailed nano-seismology. According to various scaling laws, such as the Gutenberg-Richter law, a distribution of moment magnitude against frequency of events[12], crack length and moment magnitude in a mm-sample in the LVP may be tentatively scaled up to explain the origin and mechanisms of large earthquakes. The tiny laboratory "quakes" produced by micro- to millimeter scale faulting in the sample are characteristically ultrasonic and can be detected using acoustic sensors into the ultrasonic range (e.g. 0.1 – 4 MHz). Based on the

source parameters extracted from the recorded waveforms, we can closely track event initiation, clustering and propagation, while deformation and transformation processes take place[13–17]. In this way, how fractures initiate, nucleate and propagate for intermediate and deep earthquakes can be tackled and give us important information about earthquakes by processing frequency-magnitude distribution and source parameters.

In recent years, many geophysicists started using AE techniques to investigate mineral rheology, rock failure and fault frictional behavior[16–19]. The method provides a direct way to probe the fundamental mechanical behavior of rocks under HPHT deformation conditions. Meade *et al.*[20,21] performed AE measurements with one sensor in a DAC for the phase transformation of Si and Ge. Yanbin Wang *et al.*[17] and Schubnel *et al.*[16] carried out AE measurements in a D-DIA type multi anvil press in 13BMD at APS with 6 sensors. Officer and Secco[18] used 6 sensors with walker type configurations to investigate the transformational faulting in Fe_2SiO_4. Gasc *et al.*[22] used the cubic type configuration and 6 sensors to detect the dehydration of serpentinite. Dobson *et al.*[23] used 2 sensors to investigate the dehydration of serpentine, which can only locate the position along z direction. Jung *et al.*[24] used 4 sensors in the walker type modulus at Geophysical Laboratory of Carnegie institution of Washington. Ohuchi *et al.*[25] used 6 lead-zirconate titanate sensors in cubic geometry to detect AE from brittle behavior in dunite and harzburgite due to localized heating. A large cracking event and a burst of AE was associated with a stress drop estimated from changes in the lattice micro-strains observed by synchrotron X-ray diffraction.

Because of the small size of the experimental assembly (i.e. sample) and complicated waveform travel paths, coupling precise AE monitoring with multi-anvil press remains challenging[19,20,22–24,26]. To accurately locate AEs relative to the small sample size (usually several millimeter), the hypocenter should be resolved with sub-millimeter accuracy. There are several parameters that restrict the accuracy of three-dimensional hypocenter location. The most important one is the limited number of sensors, which drastically decreases the accuracy of location and can preclude focal mechanism analysis (at least 8 sensors are needed for moment tensor analysis, whereas 6 is possible but the solution remains underdetermined, and a minimum of 4 sensors are required for 3d location)[22,27–29]. Experiments with controlled strain rate and stress are critical to simulate the earth interior, and subduction zones in particular. Synchrotron radiation can be used to track deformation processes by monitoring sample length, strain and stress, and to correlate AEs with phase transitions. When it is coupled with synchrotron, AE technique is an effective way to monitor the brittle behavior of material under HPHT. Thus, with *in situ* X-ray diffraction and imaging, we can directly correlate between dehydration reactions and onset of (micro-)cracking[3,19,21,22].

The objective of this study is to design and build a highly reliable HPHT AE system, coupled with synchrotron X-ray diffraction and imaging and controlled deformation in the LVP at the P61B beamline station. The AE detection system was previously briefly mentioned in the manuscript describing the LVP station P61B[1]. Here we elaborate on the development and performance of this system. AE detection is primarily intended for geophysical investigations on the brittle behavior of materials under HPHT and stress, with particular application to the study of the origin of intermediate and deep-focus earthquakes. In order to achieve high location accuracy and rudimentary moment tensor analysis, an assembly with six anvils and six sensors with cubic symmetry was designed. The wave speeds of the PTM and the positions of the six sensors were calibrated to provide a basis for accurate AE source location. A candidate assembly was designed to be quiet and thermally insulating to optimize AE detection of the sample with sensor temperatures on the anvils below 100 °C. To test the reliability and accuracy of this system, different types of self-designed AE sources were used to calibrate this system with and without synchrotron X-rays. The present AE setup exhibits extremely high efficiency and accuracy in detecting AE and quantifying their characteristics under HPHT.

2. Beamline instrumentation

Figure 1 shows a schematic representation of our AE system at P61B. The various components include a high-pressure system (the LVP and high-pressure assembly), high-flux white X-ray beam, X-ray diffraction detector system and radiography system, and AE detection and process system. As shown in the schematic drawing of the AE detection setup, AE waves propagate through the PTM and are detected by piezoelectric ceramic sensors at the rear of tungsten carbide anvils. The details of this setup and results of various experiments are given below.

2.1. Press operation

A Hall-type LVP called “Aster-15” (Voggenreiter mavo LPQ-1500-1000) was used to achieve HPHT and controlled deformation. The LVP is permanently installed at the synchrotron end-station P61B at PETRA III, DESY (Hamburg, Germany). This press and the AE detection system can be operated both with and without synchrotron X-rays. “Aster-15” is a cubic anvil apparatus with 6 independent hydraulic rams that can provide a combined load of 15 MN (5 MN per axis). As described by Farla *et al.*[1] the LVP supports various modes of compression for various assemblies. The ‘MA6-6’ assembly comprises 6 second-stage tungsten carbide anvils compressed by 6 steel first-stage anvils in the LVP and supports both isotropic and anisotropic compression modes, whereas the MA6-8 assembly featuring 8 second-stage tungsten

carbide anvils only supports isotropic compression, but can be used to reach ultra-high (30 GPa) pressures. For controlled deformation experiments described in this work, we only use the MA6-6 assembly.

Currently, the following anvils are available and compatible with the AE setup: 15 mm TEL (0.1-2 GPa), 12 mm TEL (0.5-4 GPa), and 9 mm TEL (up to 7 GPa), where TEL = truncation edge length and pressure range given in brackets. At the moment, only one X-ray transparent anvil (c-BN) is available for *in situ* X-ray diffraction with a TEL of 12 mm (Fig. 2c). Furthermore, a third stage of smaller WC and sintered diamond anvils can be tentatively used with AE detection; however, it was not tested until now. The third-stage assembly permits access to higher pressures with smaller TELs of 5 mm (2-10 GPa), 4 mm (up to 15 GPa) and 2.5 mm, (up to 18 GPa); although tolerable stress is unknown. Using the MA6-6 compression geometry, the sample in a cubic PTM (Fig. 2a) can be directly subjected to a differential stress from the advancement of a pair of opposed hydraulic rams in the LVP. The maximum load that the anvils, 35 mm in diameter, can withstand is greatly reduced by a WC support ring behind each anvil that protects the piezoelectric sensor for AE detection. For this reason, the maximum sample pressure on each anvil type with given TEL may appear to be on the low side.

Since the six hydraulic rams of Aster-15 can be controlled independently, 1 or even 2 pairs of opposite hydraulic rams can be advanced or retracted at steps of constant displacement rate for anisotropic compression operation. Typically, in a short period (less than 15 – 20 min), the strain rate of the sample in the PTM becomes directly proportional to the displacement rate of the rams. Hence, it is possible to control the strain rate in the sample at HPHT by changing the ram displacement rate, facilitated by the alumina pistons inside the assembly that transmit the stress and strain directly to the sample. Note that crushable alumina is sometimes added to accommodate some of the initial compaction of the assembly during cold compression without immediately causing stress on the sample (Fig. 2a). Once at high pressure, the sample is typically deformed using the vertical rams (No1 and No2) at a controlled displacement rate (µm/min), which shortens the sample along the axial direction. Sample shortening can be directly imaged through the anvil gap of the MA6-6 assembly by X-ray radiography. The sample strain is expressed as the change in axial length ΔL with original length L, $\varepsilon_s = -\Delta L/L$, often referred to as the standard or engineering strain. For an experiment using X-ray radiography to measure the sample strain history, the true strain expression $\varepsilon_T = -\ln(\Delta L/L)$ may be preferred. Note, the negative sign for compression. Although, less applicable to brittle deformation AE experiments, it is important to realise that the displacement rate is also proportional to stress σ^n in the sample, where n is the stress exponent (e.g. 1 for Newtonian creep and 3-5 for power-law, non-Newtonian creep). The relationship between

155 strain rate and stress is called a flow law, which is applied to plastic deformation in materials, particularly
156 to mineral assemblages in the deep Earth to understand convective plate tectonics.

157 After several experiments, e.g. with TEL = 12 mm anvils in this study, it is possible to predict the PTM
158 cube size and gasket thickness between each pair of anvils for any press load and, additionally, for the
159 advancement of a pair of rams during anisotropic compression (Fig. S1). This is done using the logged
160 data of the ram displacements by the press control software. The cube size can be calculated from the
161 reference 'touch' point when the rams contacted the assembly with second-stage anvils and cubic
162 pressure medium in between. The touch point is defined as the first match between the programmed
163 load profile and the actual load on the master ram. This method of estimating the cube size under
164 pressure was confirmed by imaging the anvil-to-anvil distance using X-ray radiography. It is therefore
165 possible, also without using synchrotron X-rays, to determine the distance between the AE sensors on
166 the anvils at any press load during an experiment, which improves the accuracy of AE event location.

167

168 ## 2.2. Setup using synchrotron X-rays

169 X-ray techniques can be crucial to an *in-situ* AE deformation experiment (Fig. 1). Imaging is done using X-
170 ray radiography. The white-beam X-ray microscope at P61B captures a part of the white beam that
171 passed through the sample. A small fraction of the polychromatic X-rays, converted into visible light
172 using a GGG:Eu scintillator, is observed by the PCO.edge 5.5MP sCMOS camera. As mentioned earlier,
173 just like the anvil to anvil distance can be imaged, the length of sample along axial direction can be
174 determined with µm precision during the various stages of a typical AE deformation experiment (i.e. for
175 the cold compression, heating and deformation). Because the beam size is usually smaller than sample
176 length, scripted up / down movement of the press Z-stage can be performed with fixed steps and choice
177 of image overlap (e.g. 30%). After acquisition for a given displacement, the beamline software stitches
178 the frames together into a single montage, which can be processed to estimate the length of the large
179 sample. During anisotropic compression, many such acquisitions will be carried out, in order to obtain
180 the final macro-strain and strain rate history of the deformed sample, and the total sample strain. The
181 sample image data may also give information to the onset of cracking in the sample, which may coincide
182 with the onset of AE production. However, due to various limitations, only large crack interfaces can be
183 seen. Subsequent micro-tomography on the recovered sample may be performed elsewhere to obtain
184 additional information of the micro-crack networks that developed during deformation, assuming
185 decompression did not modify the cracked sample.

Diffraction patterns, in the 30-160 keV range, can be collected during the different experiment stages using two high-purity solid state germanium detectors (Ge-SSD) by Mirion (Fig. 1). One detector is tilted at a given scattering angle (2θ) in the vertical plane (azimuth angle = 0°) to collect the diffracted X-rays from the anvil gap (for MA6-6). A second detector is positioned at the same scattering angle, but in the horizontal orientation (azimuth angle = 90°). To receive the diffracted X-rays from the sample at 90° azimuth, one of WC anvils must be replaced by an X-ray transparent anvil, such as c-BN. One Ge-SSD is sufficient if only changes in the sample composition are monitored during the HPHT deformation experiment. Particularly, one may be interested in embrittlement processes resulting from phase transformations or dehydration/decomposition of hydrous phases. If the onset of AE production matches the observed change in the sample by X-ray diffraction, a scientific case can be made. Additionally, the detector can be used to capture diffraction patterns of a pressure-temperature (PT) marker (a combination of two materials with well-known equation of states), to estimate the pressure and temperature *in situ*. However, this typically does not work well once the sample and PT marker are subjected to a deviatoric stress. A second Ge-SSD can be used in combination with the other one to estimate stress in the sample (or piston) from changes in the lattice micro-strain recorded in the diffraction patterns from both detectors at 0° and 90° azimuth[30]. This is only possible for controlled axisymmetric compression due to the lack of azimuthal detector coverage at the moment. Notwithstanding, semi-quantitatively, a sudden drop in stress during deformation can be correlated to a burst of AE activity.

Although this study does not expand its scope towards employing energy-dispersive X-ray diffraction (ED-XRD) techniques and the use of geological samples, which may be candidates responsible for earthquakes in the Earth, we point out these techniques, because it is already possible to carry out such experimental research at P61B.

2.3. A high-pressure assembly for AE detection

For an AE experiment in the LVP, we are mainly concerned with the need to design a cool, low-noise assembly. The assembly requires good thermal insulation, so that the sensors attached to the back of the anvils do not significantly heat up beyond 100 °C while the sample is at much higher temperature.

We pressed boron-epoxy cubes (amorphous boron powder and epoxy, 4:1 by weight) with 16 mm edge-length as PTM. A boron and epoxy ratio by weight of 8:1 was also attempted. However, due to low epoxy ratio, the cube becomes too brittle, generating many cracks (and AE) during an experiment. Until now, only the 12 mm TEL anvils have been used in combination with a 16x16x16 mm^3 pressure medium

with a maximum sample size of 6 mm in length and 4 mm in diameter, and routinely a sample size of 5 mm in length and 2.5 mm in diameter. The available pressure range for this anvil/cell assembly is 0.5 to 4 GPa. Electrical resistive heating is performed using a graphite tube furnace inside the assembly (Fig. 2a). Six anvils, each with sensor and support ring, are inserted into an aluminium alignment frame (Fig. 2b, c and Fig. S2c, d) to compress a cubic cell assembly (Fig. 2d). Heat was generated with a DC power supply which can be controlled both manually and automatically. A traditional assembly uses 1 mm diameter Mo electrodes in ZrO_2 insulation. However, this rod acts as a piston during controlled deformation and will easily punch through a molybdenum foil connected to the graphite heater, quenching the experiment. In order to avoid this problem, we designed a new kind of electrode shown in Figure S3a. We machined a star-shaped molybdenum foil and then wrapped it around the zirconia as shown in the Figure S3b. In this way, abrupt quenching can be avoided during controlled HPHT deformation. In addition, this electrode design further improves thermal insulation.

To generate high stress in the sample, we placed dense and crushable Al_2O_3 (alumina) pistons on both ends of the sample. The diameter and length of sample can be changed with different strain or strain rate. The pressure-load relationship was previously calibrated at room temperature by the abrupt electrical conductivity changes induced by the Bi phase transitions (2.55 GPa, 2.7 GPa and 7.7 GPa) [31]. AEs signals were recorded during each experiment, including the cold compression, heating, deformation, quenching and decompression stages. During the cold compression procedure, especially pre-pressing, many AEs signals were recorded which originate from the crushing of the boron-epoxy cube and crushing of the gaskets. When oil pressure in the master ram is higher than 30 bar (16/12 assembly), no further AEs signals were detected, demonstrating that our AE assembly is quiet under HPHT conditions. This feature ensures that the recorded AEs signals during holding and deformation procedure did not originate from compaction of the HP assembly. According to previous reports of Gasc *et al.*[22], except for the crushable alumina, deformation of the assembly materials will not produce a significant amount of AEs. Hence, most if not all, AEs detected should come from the brittle sample inside.

3. The AE detection system

The AE detection system for *in situ* studies of fracture and brittle behaviour of materials at elevated pressures (and temperatures) was installed and commissioned in 2020 (Fig. 1). Below is a description of the purchased hardware and software provided by GMA/MISTRAS (Physical Acoustics Corporation, PAC in the USA).

3.1. Hardware and software

This AE detection system operates with the following specifications. A 6-channel system with 3x PCI-2 boards is built into a PC system. The bandwidth range is 0.1 to 3 MHz with a sampling rate up to 40 MHz (25 ns interval) and a resolution of 18 bit A/D. AEwin software for 6-channel data acquisition replays function with waveform recording and 3d location algorithm included (first threshold crossing, FTC). In addition, there are 6x 20/40/60 dB preamplifiers with internal filters to amplify the original signals. The preamplifiers are connected to the PC by 30 m long low-attenuation co-axial cables from the experimental hutch to the control hutch. Each signal from the sensors are converted to an electrical signal and sent to the amplifiers by low impedance coaxial cables. The amplified AEs are sent to six channels PCI-2 boards, which transform the analogue electrical signals to digital signals. There is a possibility to set up additional analogue and front-end filters (Table S1). During the experiment, the preamplifiers are mounted on the side of the hydraulic rams (Fig. 2d), with No1 and No2 covered with 3 mm thick lead plating to avoid damage by scattered X-ray radiation. The AEwin software allows data acquisition and real-time analysis of waveforms. Furthermore, the AEwin software automatically calculates all AE features of the waveform of each hit, which include MARSE (Measured Area under the Rectified Signal Envelope) energy, counts, duration, amplitude, peak and average frequency, and so on (Fig. 3b).

For high-pressure AE experiments in the LVP, the sample is surrounded by various components of the cell assembly (Fig. 2a) and pre-formed gaskets, which produce significant noise when crushed between the anvils. Therefore, in each channel a threshold above the background noise is critical to ensure no signals are missed, while each sensor does not trigger on the noise either. Tests have shown that the Aster-15 LVP generates surprisingly low-amplitude noise when operating. The AEwin software, monitoring the AE sensors, could be reliably configured to trigger with a first threshold crossing (FTC) at 28 – 30 dB using 40 dB gain pre-amplifiers. In other words, according to the relationship: dB_{AE} = 20 log (U/Ur) – P, where U is the threshold in µV, Ur (1 µV) is the reference voltage, and P is the preamplifier gain (dB), the first threshold crossing using the AE setup at P61B can be as low as 0.002 V.

We exclusively used trigger-based AE hit acquisition and waveform saving to collect data. Continuous waveform streaming is also supported, but lowers the maximum sampling rate on the other triggered channels. Since the system is continuously monitoring AE activity, waveform data of each hit before the first threshold crossing are also recorded. AEwin can be used to constrain events (collection of 6 nearly-simultaneous hits) and then be used to export the associated waveforms for further analysis. Alternatively, all raw data can be exported for data analysis from scratch by alternative software.

However, the second option is generally not necessary. For accurate event source location, determination of the first P wave arrivals is critical. More advanced crack mechanism analysis is possible when the sensitivity of each sensor is precisely calibrated (Fig. S4). Once the amplitudes of the sensors are calibrated, information of the integrated area and sign (+ or -) of the first arrivals can be used to carry out moment tensor inversion analysis to obtain the focal mechanisms. Even so, with a limited number of 6 sensors, moment tensor inversion is underdetermined and any result should be scrutinized with utmost caution.

3.2. Sensors

Broad-bandwidth, high-frequency sensors, MICRO-200HF, also from GMA Company are chosen for the calibration experiments in this study. The Micro200HF sensor has good frequency response in the range 500 – 4000 kHz with a resonant frequency around 2500 kHz (Fig. S4). An ultrasonic broadband sensor is suitable here to characterise the possible range of frequencies generated by fractures propagating over distances ranging from micrometre to millimetre (typical range for grain size to sample sizes). Small size (9.5 mm in diameter and 11 mm in height) and wide temperature range (-65-177 ℃) features make this sensor an ideal candidate for application on samples deformed at HPHT in the LVP. While expensive, these sensors are reusable. However, several stages of development were conducted to ensure the sensor on the back of each anvil does not break or detach during an experiment. One of the main challenges is the survival of all sensors throughout successive experiments, particularly during beam time. The replacement of a sensor on an anvil is time-consuming and each new sensor must first be calibrated before using.

The AE setup was optimized during the course of this study. In principle, the six sensors need to be protected from anvil compression by slightly taller tungsten carbide supporting rings (12 mm) behind the 35 mm diameter tungsten carbide anvils, with the sensor in the middle. For the first set of experiments until HH411, shown in Table 1, the assembly was continuously improved based on an initial design shown in Figure S2a, c, e. The optimised design, as confirmed by a high sensor survival rate after experiment HH411 and subsequent experiments (Table 1, Fig. 2, S2b, d, f), comprises a thin foil of Cu separating the support ring from WC anvil, which are held together by a surrounding brass ring with small screws in the corners. Each WC support ring is grooved to feed through the cable that connects the sensor to its amplifier. Each of the 6 sensors is attached to the back of an anvil using a special high-temperature (200 ℃) resistant adhesive (cyanoacrylate), which ensures that each sensor does not detach from its anvil during an HPHT experiment. Lastly, all anvils with attached sensors are self-aligned in a frame machined

from a single block of aluminium (Fig. 2c). Since the sensitivity of the AE sensors must be similar, the sensitivities are checked before each experiment to ensure AEs can be detected consistently. In addition, a full calibration can also be performed, as shown in Fig. S4.

3.3. Locations of AE events

The locations of the six AE sensors are symmetric with respect to the centre of sample, which constraints the location of AE source. Consequently, when the hypocentres of cracks that occur closer to the centre (0, 0, 0), the first P-wave arrival times will be very similar. The three-dimensional sensor arrangement allows 3-d location of AE sources from the time difference, Δt, between the arrivals at each opposite sensor (Fig. 3a). AEwin can do all the hit detection by FTC, AE and event characterisation in real time. Following an experiment, the data can be replayed in AEwin to analyse them in detail, and can be exported for further processing e.g. using MATLAB.

We briefly explain the operation of AEwin for 3d event location. For more information, please refer to the software manual provided by GMA/MISTRAS or PAC. AEwin requires some initial information on the positions of the sensors as well as the average wave speed of the AEs through the medium (in our case, the WC anvil and B-epoxy PTM). The AE sensors are labelled with S1-S6 as shown in Figure 3c, d, corresponding to the ram numbering of Aster-15. For a typical experiment with a B-epoxy cube size of 15 mm under pressure (see Table 1, Fig. S1) and anvil length of 37 mm, the sensor x, y, z coordinates in mm are: S1 (0, 0, 44.5), S2 (0, 0, -44.5), S3 (44.5, 0, 0), S4 (-44.5, 0, 0), S5 (0, 44.5, 0), S6 (0, -44.5, 0). Furthermore, the average wave speed for the 16/12 assembly with B-epoxy PTM was determined as 5522 m/s (Fig. S5), based on sensor pulse tests carried out under load for the experiments listed in Table 1. Since the triggered waveforms can be exported for reanalysis, manually changing the sensor positions in the software is not needed when increasing the load or starting controlled deformation by advancement of a pair of rams.

Fundamentally, the theory of AE location is a simple time-distance relationship given by the average velocity of sound waves. Each arrival time difference implies a difference in distance between two opposite pairs of sensors. The 3-d location of AE events using multiple sensors (minimum requirement is 4) is calculated in AEwin using a modified multi-regression analysis algorithm, which is not revealed. However, the manual describes the basic computation as follows:

$$\Delta t_{i,j}=\left[\sqrt{(x_i-x_s)^2+(y_i-y_s)^2+(z_i-z_s)^2}-\sqrt{(x_j-x_s)^2+(y_j-y_s)^2+(z_j-z_s)^2}\right]/v$$

$$\chi^2 = \sum_{i=1}^{6} \left(\Delta t_{i,obs} - \Delta t_{i,calc} \right)^2 \quad (1)$$

where x_i, y_i and z_i are the coordinates of sensor i, x_s, y_s and z_s are the source position, Δt is the difference in the arrival time of the signal on sensor i and sensor j. For sub-mm accuracy on the localisation of events in 3d space, the survival of all six sensors is very important. Using 5 sensors may still give decent results, but is less optimal. Using the minimum required number of 4 sensors does not permit accurate 3-d location of events.

This study offers MATLAB scripts (**recent versions in Github**) for reprocessing the triggered waveforms of events established by AEwin. This is helpful because each hit associated with an event in AEwin is also characterised by MARSE energy, number of counts, amplitude, (peak) frequency, and so forth. Hence, each event can be associated with a particular AE feature. The MATLAB script takes advantage of these exported data to reprocess and plot the 3d location of events and their AE features for each experiment. All data presented in this paper are reprocessed by these scripts. An example of the script and an explanation of the code can be found in the Supplementary Text. Note, the MATLAB script borrows 3d location code by Li (2021)[32].

In summary, many previous studies used only two or four sensors, which is only useful for roughly locating events between two opposite sensors, but not for true 3d location. Here, we present an optimised AE assembly, which offers excellent reliability of all six sensors working during successive experiments. Furthermore, these commercial sensors are re-calibrated in-house after being glued on the back of anvils (Fig. S4), which is critical for accurate 3d location with sub-mm precision without resorting to using seismic algorithms, and may be useful for rudimentary moment tensor analysis.

3.4. Temperature calibration

Internal temperatures of various test assemblies were determined by establishing temperature-power (P-T) relationships before a normal experiment. The external anvil temperature was also monitored. We used a C-type ($W_{90}Re_5$-$W_{74}Re_{26}$) thermocouple for these tests. From several repeated experiments, the estimated temperature uncertainty is around ±50 K. First, we used the assembly shown in Figure 4a with graphite heater of 6 mm and 4 mm outer diameter and the P-T relationship is shown in Figure 4a. The assembly with a smaller heater has higher heating efficiency. Figure 4b shows anvil temperature using a 6 mm heater in an assembly and the temperature reaches more than 100 °C when the central chamber temperature is just 500 °C. High anvil temperature reduces the sensitivity of the piezo-sensor, and the

cyano-acrylate becomes soft leading to the detachment of the sensor from the anvil. The anvil temperature using a 4 mm diameter heater is much lower than using a 6 mm diameter heater, which is good for efficiency and survivability of the sensor.

4. Experimental results

4.1. Managing ambient AE noise

A major issue in studying AE at high pressure and temperature is the noise of the whole system, especially the crushing of the high-pressure assembly. To achieve low AE rates and generally a quiet PTM, we tested pre-fired pyrophyllite and boron epoxy (with a boron to epoxy ratio of 4:1) as PTM. A typical AE deformation experiment (Table 1) consists of three different stages: 1. Compression to target pressure (with rate about 12.5 MPa/min). 2. Deformation (typically at -10 µm/min per hydraulic ram, results in approx. 2×10^{-5} s^{-1} in a 4 mm sample) 3. Decompression to room conditions (with rate about 6.25 MPa/min).

The different behaviour of boron epoxy (HH413) and pyrophyllite (HH420) was compared as function of oil pressure and time (Fig. 5). During the start of compression up to about 30 bar oil pressure, and at the end of decompression, we collect many AEs. The AEs obtained during the initial compression stage come from the initial crushing of the cube and gaskets, while the AEs at the end of decompression come from the breaking apart of the cube due to stress release. We can see that the boron epoxy PTM and preformed gaskets are somewhat quieter than pyrophyllite (Fig. 5, Table 1). In addition, boron-epoxy cubes are extremely quiet during the deformation procedure, which ensures that many, if not most, AEs during deformation must come from the sample. Based on the above result, and the fact that boron-epoxy is much more X-ray transparent, we chose boron-epoxy (4:1) as the PTM for all test experiments (Table 1).

The noise of the LVP Aster-15 was also monitored. As previously discussed, the FTC for triggered AE detection can be set to 28 – 30 dB (0.002 V with a 40 dB preamplifier gain). This is an exceptionally good result in comparison to previous studies (e.g. Gasc *et al.*, Officer and Secco[22,33]). An example of the background noise level in the LVP at high pressure load is given in Figure 6. Notably, the FTC at 0.002 V is shown in the magnified portion of the waveforms for all six sensors. While the FTC method does not truly pick the first P-wave arrival as seismic algorithms do, the accuracy is extremely high. For the event shown in Figure 6, the arrival times are nearly simultaneous on all six sensors, which suggests the crack must have taken place close to the sample centre (i.e. at the centre of the assembly). Furthermore, the waveforms here are interpolated with 2 ns interval from the original data with 25 ns interval (40 MS/s)

prior to P-wave arrival picking (Table S1). We believe this method offers the best possible accuracy on the 3d location determination of events.

4.2. Experimental runs without X-rays

In order to test the accuracy of our AE detection setup, several experiments were run with different self-designed AE sources, which are tabulated in Table 1. For most tests, the compression, deformation and decompression stages were carried out, and AEs were monitored during each stage. The locations of events were determined by our own MATLAB script (see Supplemental Text) based on the different arrival times obtained by the FTC method, selected just above the noise (i.e. at 30 dB). The locations of AEs of all experiments are shown in Figure 7. The circle size indicates the MARSE energy of an AE event. The gray area indicates the position and size of original AE source (i.e. sample). The amplitude distribution of each channel is shown by a histogram on the right side. For some experiments, a blank histogram is shown, which indicates the corresponding AE sensor was not working (HH308 and HH321). The optical images of the AE source after an experiment (where recoverable) are shown in the insets for each experiment (Fig. 7). Next, we describe each experiment in more detail.

First, we used glass beads with different diameters (≥200 µm in HH287, ≤100 µm in HH289) wrapped in Ni foil as AE sources. The AE location results are shown in Figure 7a, b. The located AEs outline the shape of the glass beads sample. It appears that the MARSE energy of HH287 is higher than that of HH289, consistently with larger cracks forming in the larger glass beads.

The results from calibration experiments using 1) fused glass wedge (FGW) pistons (Fig. 7c), 2) FGW pistons + small glass beads (Fig. 7d), and 3) saw-cut alumina pistons (Fig. 7e) all show the original outline of the saw-cut at 30° to the compression direction. With the serrated sliding surfaces of both FGW pistons in contact, numerous AEs were produced during the mutual movement of the FGW. It is clear that the FGW + small glass beads produce the most AEs, due to cracking of the additional glass beads. In contrast, with the addition of an h-BN soft layer between the FGW, the pistons barely produced AE and only a few cracks were detected despite obvious FGW piston displacements. The located source distribution is also compatible with the slope of the FGW, suggesting high location accuracy (around 0.5 mm) of our AE detection setup. To rule out the possibility that AEs come from the breaking of the FGW body, we repeated an experiment with saw-cut solid molybdenum wedges with small glass beads glued to the sliding surfaces, as shown in Figure 7g. We observe nearly the same location configuration as in the other experiments using brittle FGW.

The AEs source locations should be homogeneously distributed in the glass beads sample. However, they are typically plotted with a "star-shape" artefact, which appears to be a result of the MATLAB minimisation algorithm "fminsearch" based on setting the tolerances for the function (TolFun) and step size (TolX) too low. That means the solver can fail to recognize when it has converged, and can continue futile iterations. At the moment, we have not been able to remove this artefact, only slightly reduce it (not presented here). Microstructural analysis did not reveal any visible cracks along the 'star-shaped' 45° directions in the recovered glassy beads samples and pistons. In most cases, the overall shape of each AEs cluster is reasonably in agreement with the sample volume in each experiment (Fig. 7).

To further test the location accuracy of calculated AEs, we performed a two-layer experiment with two AE sources separated by a soft h-BN block, as shown in Figure 7f. As expected, we accurately identified the positions of the two layers of AEs sources in the PTM cube. The distance between two clusters of AE sources match the location of the two glass beads layers in the assembly, again suggesting the high accuracy of our HPHT AE system. Based on these test runs, we can say that our AE detection system at P61B offers high accuracy (~0.5 mm) and can be used to detect AE under HPHT for scientific problems, such as the origin of earthquakes in the Earth's mantle.

4.3. Experimental runs with X-rays

To test the ability of our high-pressure AE system coupled with X-ray radiography and ED-XRD, we conducted two additional AE experiments using the wiggler white-beam of P61B. In the first experiment, BT540, we again used glass beads in Ni foil (25 µm foil thickness) as AE sources. Montages of multiple X-ray radiography images, making up the total sample length were obtained by X-ray radiography at several selected deformation points (Fig. S6a). As indicated by the two arrows, we can see that the length of sample gradually decreases with increasing strain due to the constant displacement rate of the hydraulic rams during anisotropic compression. The strain history of the glass beads sample at different time points is summarised in Figure S6b. The strain history is shown by the red line, which indicates a constant strain rate. The deformation time was 45 mins using rams No1 and No2 at a constant displacement rate of -10 µm/min per hydraulic ram. The final strain, ε, is around 9.3% with a strain rate, $\dot{\varepsilon}$, of 3.3×10^{-5} s^{-1}. The AEs were recorded and locations (n=1275) are shown in Figure 7h. We also provide additional information on the typical peak frequency (around 550 kHz) produced by the cracking of the glassy beads in this experiment (BT540) (Fig. S7), which provides some reference for other experiments. The high resolution of the X-ray radiography images allows accurate *in situ* determination of the strain and strain rate in a sample deformed over one or multiple steps, by controlling the ram displacement

rates. Hence, the deformation process can be precisely tracked *in situ*, also on other geological materials of interest, which may produce AE.

Normally two WC anvils on either side of the MA6-6 assembly absorb all diffracted X-rays, which hampers the ability to estimate stress in the sample without at least two point-detectors azimuthally positioned at 0° and 90° on the diffraction cone of the sample source. A second experiment, BT587, again using a glass beads sample wrapped in Ni foil, was carried out to test the feasibility of using an X-ray transparent anvil to perform simultaneous ED-XRD, not just from the anvil gap, but on the side where a 2nd Ge-detector is placed. This special anvil is made of c-BN, as shown by the black anvil in Figures 2c and 8d. The sample was first compressed to a target pressure around 1.0 GPa (45 bar), then the sample was heated to around 700 °C and was deformed for 1 h with a total of 1 mm combined ram displacement. The initial sample length was 3.05 mm and was shortened 0.5 mm by cold compression, and a further ~1.0 mm by controlled ram displacement to around 1.49 mm (Fig. 8a). Hence, the total strain is ~50%. Through the cBN anvil, we successfully acquired the diffracted X-rays of the Al_2O_3 piston and also from the sample at high temperature (~700 °C) in detector unit 2 during deformation. The diffraction patterns of the Al_2O_3 piston and glass beads sample at 700 °C are shown in Figure 8b, c, at a 2θ of 4.02°. XRD confirms the glass beads transformed to quartz, clinopyroxene and other possible phases at high temperature. The additional phases, other than SiO_2, can be explained by the fact that the beads are unwashed (i.e. 'dirty'). Despite strong deformation and crystallization at high temperature, the round shapes of some large beads remain visible in the X-ray radiography images.

The AEs were recorded during the whole experiment procedure. However, there was no sensor on the c-BN anvil for this test and regrettably sensor No1 malfunctioned. The 3d locations of AEs obtained using 4 sensors are too unreliable and are not shown here. We confirm the usual high AE activity during cold compression. However, during deformation at high temperature, very few additional AEs were recorded. Likely, after crystallization, SiO_2 exhibits highly dense feature because the majority of glass beads break. It deforms by cataclasis and further comminution of grains, which is hard to detect. The same behavior at high temperature was previously observed by Gasc et al[22].

4.4. SEM analysis of AE sources

Various deformed source specimens were recovered and imaged at the DESY Nano-lab using a scanning electron microscope (SEM, Tescan Amber X scanning electron microscope). Briefly, the samples were prepared as follows. After the experiments, the quenched boron-epoxy cubes were cut in half, parallel to the compression axis. Figure 9a, c, e, f shows the low-magnification images of the recovered glass beads

(HH289), fused glass wedge (HH264) and two layers of glass beads (HH367). In the glass beads sample, we can observe that some porosity reduction occurred by the crushing during compression and deformation procedure. Some glass beads were completely crushed to small pieces, while others preserved their original spherical shape (as shown by the arrows). We confirm this observation for all experiments with glassy beads samples. As shown in Figure 9h, dense alumina wedge (HH321) and Figure 9i fused glass wedge (HH308), the sawtooth pattern was crushed with many cracks. The tips of the saw-cut pistons were crushed to fine powder around the sliding surface. Other parts of the piston did not crack seriously. The Mo pistons are noticeably uncracked (Fig. 7g), as expected. Hence, the only source of AE activity for HH411 could have come from the glass beads, which were glued on the sliding surfaces. All these features observed here are consistent with our AEs location results, and this confirms the high accuracy of our AE system to detect the brittle behavior of materials under HPHT.

5. Conclusion and outlook

We developed an Acoustic Emission (AE) detection setup compatible with *in-situ* X-ray imaging and diffraction at the high-energy wiggler end-station P61B LVP. The system by GMA/Mistras can record triggered AE activity as well as stream full waveforms emanating from samples at HPHT conditions, deformed by anisotropic compression in the Aster-15 LVP. This setup is a powerful tool to investigate AEs under HPHT and couple these acoustic measurements with high flux X-ray diffraction and imaging. We calibrated the wave velocity of the pressure transmission medium, which provides a basis for accurately locating the acoustic emission sources in the sample. In order to achieve deformation under HPHT with low noise, we designed a new assembly for AE experiments. The efficiency of this AE setup was confirmed by the compression and deformation of various standard samples. Some samples were designed specifically to simulate a fault gouge. The hypocenters of the AEs obtained in these experiments successfully allow imaging the fault plane. This AE setup therefore proved to be extremely effective and is available for users with and without the availability of synchrotron X-rays at the P61B LVP station. We expect scientifically pioneering studies on (Earth) materials under high pressure, temperature, strain rate, and stress, to investigate processes such as dehydration embrittlement and phase transformations in the Earth from which intermediate and deep earthquakes may originate. We offer thus a powerful tool to investigate mineral and material rheology at HPHT.

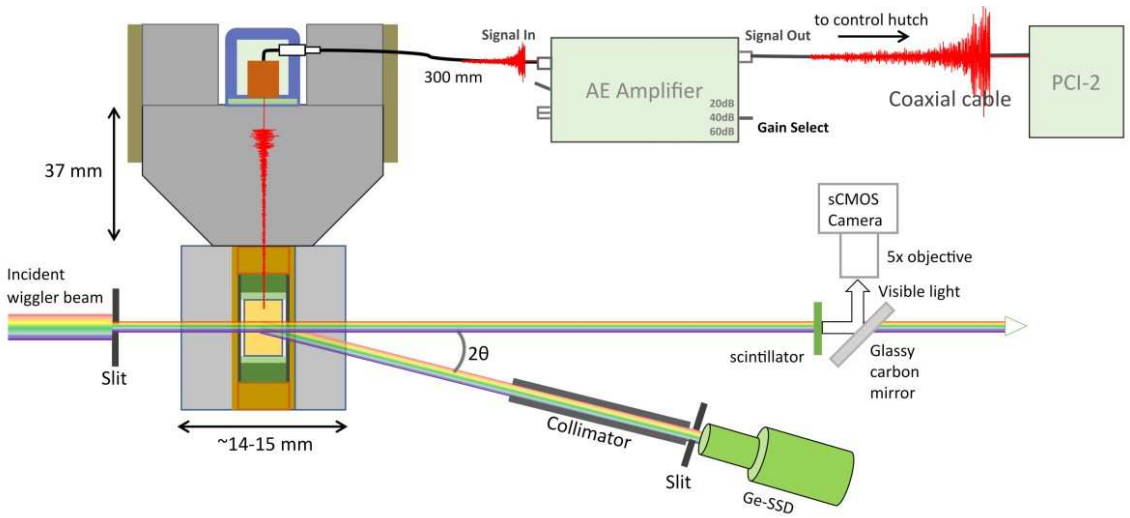


Figure 1. The schematic diagram of a portion of the MA6-6 assembly, acoustic emission system and the X-ray beam path for imaging and diffraction at P61B. The upper left shows the tungsten anvils and supporting ring. The sensor is connected to AE amplifier and PCI-2 with coaxial cable. The lower part shows the high-pressure assembly with the white X-ray beam for imaging and X-ray diffraction.

Figure 2. Photographs of the HPHT assembly used to conduct AE experiments in this study. (a) The cell assembly includes components such as the Boron-epoxy gasket, zirconia sleeve, crushable alumina piston, h-BN capsule, Sample (glass beads, glass rod, or other sample), graphite heater, dense alumina piston and cubic boron-epoxy PTM from left to right side. The electrodes are not shown in here and can be found in Figure S4. The schematic drawing of the assembly is shown below. (b) The brass supporting ring and tungsten carbide anvil and anvils after assembly (left). The anvils have a truncation of 12 mm. Six frames and anvils with gaskets after assembly (right). (c) The side view (left) and top view (right) of the whole assembly that include the alignment frame and anvils. (d) A photo of the LVP with amplifiers on each ram and the whole assembly under press load.

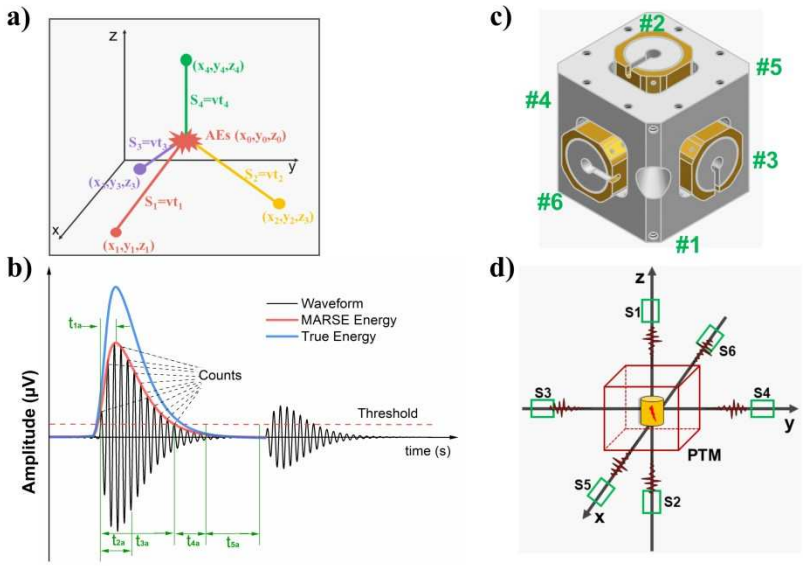


Figure 3. The main parameters that were used to determine the features of a waveform. (a) Three-dimensional location schematic picture. (b) the threshold based hit detection method to detect a transient hit and features used to identify a hit, where t_1 denotes rise-time (from threshold to peak amplitude); t_2 denotes peak definition time (PDT); t_3 means duration time, t_4 denotes hit definition time (HDT); t_5 denotes high lockout time (HLT); (c) The illustration of anvils and sensors location. (d) Three-dimensional location of six sensors (s1 to s6), pressure medium (red box) and sample (yellow cylinder).

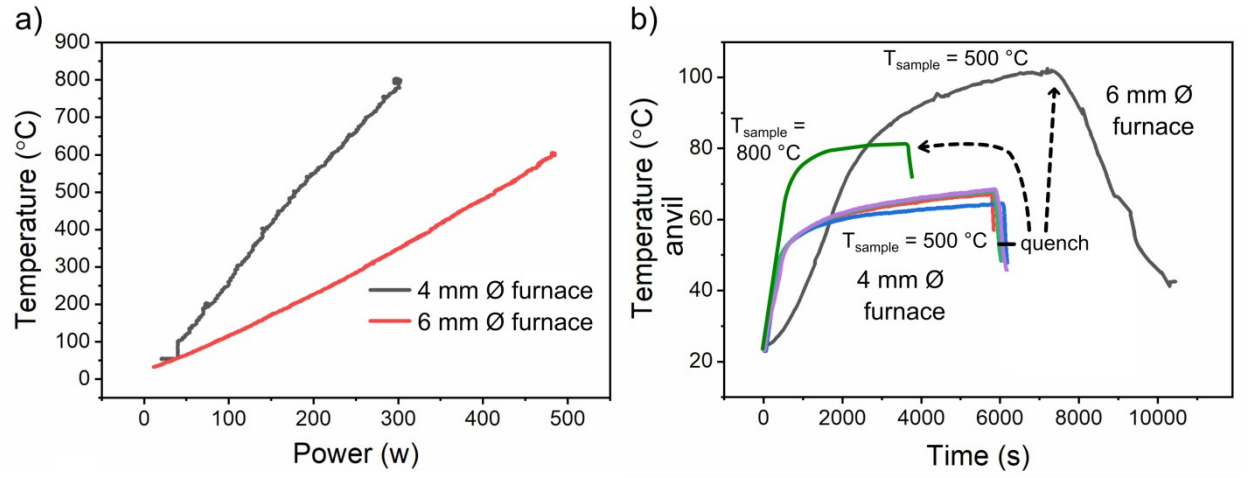
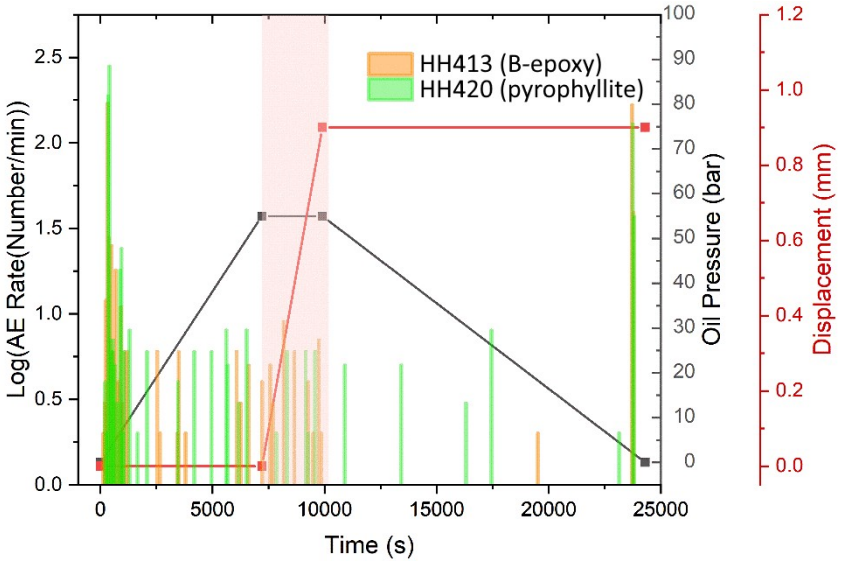
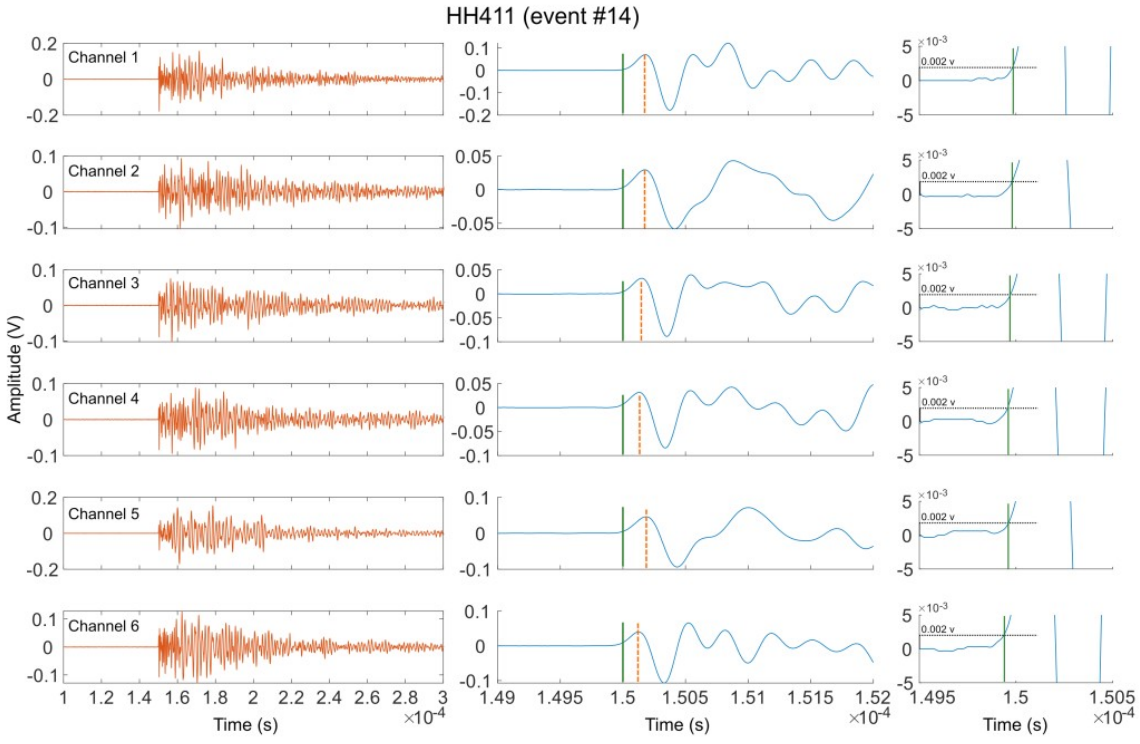


Figure 4. (a) The chamber temperature as function of power for assembly with graphite heater of 6 mm and 4 mm. The heater wall thickness is 0.5 mm in both cases. (b) The evolution of the anvil temperature over time when the sample in the assembly is heated to temperatures of 500 °C and 800 °C.



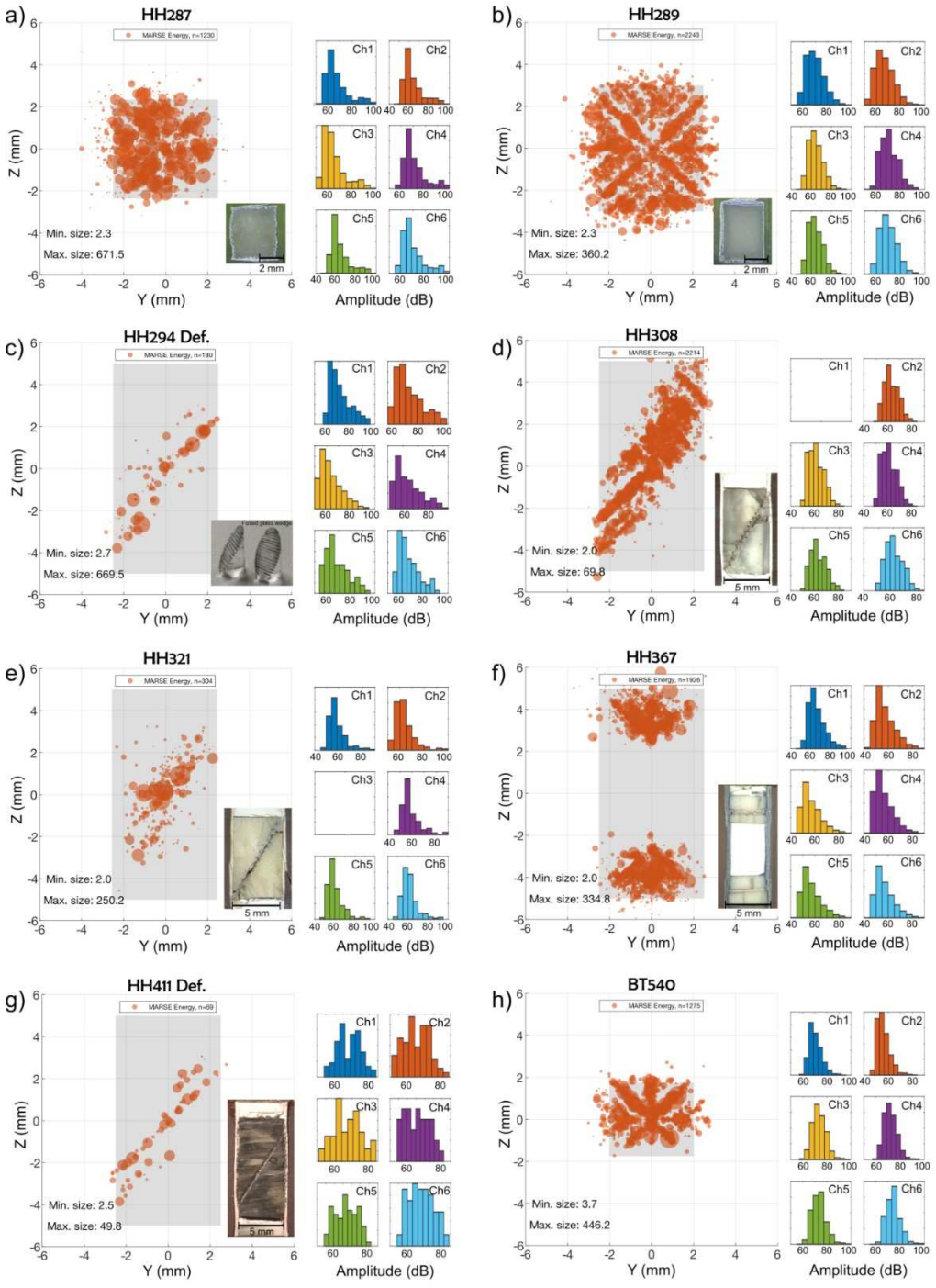
555

556 **Figure 5.** The logarithm of acoustic emission hit rates, oil pressure, and displacement as a function of
557 time during the typical experiment procedure with boron epoxy and pyrophyllite as PTM.



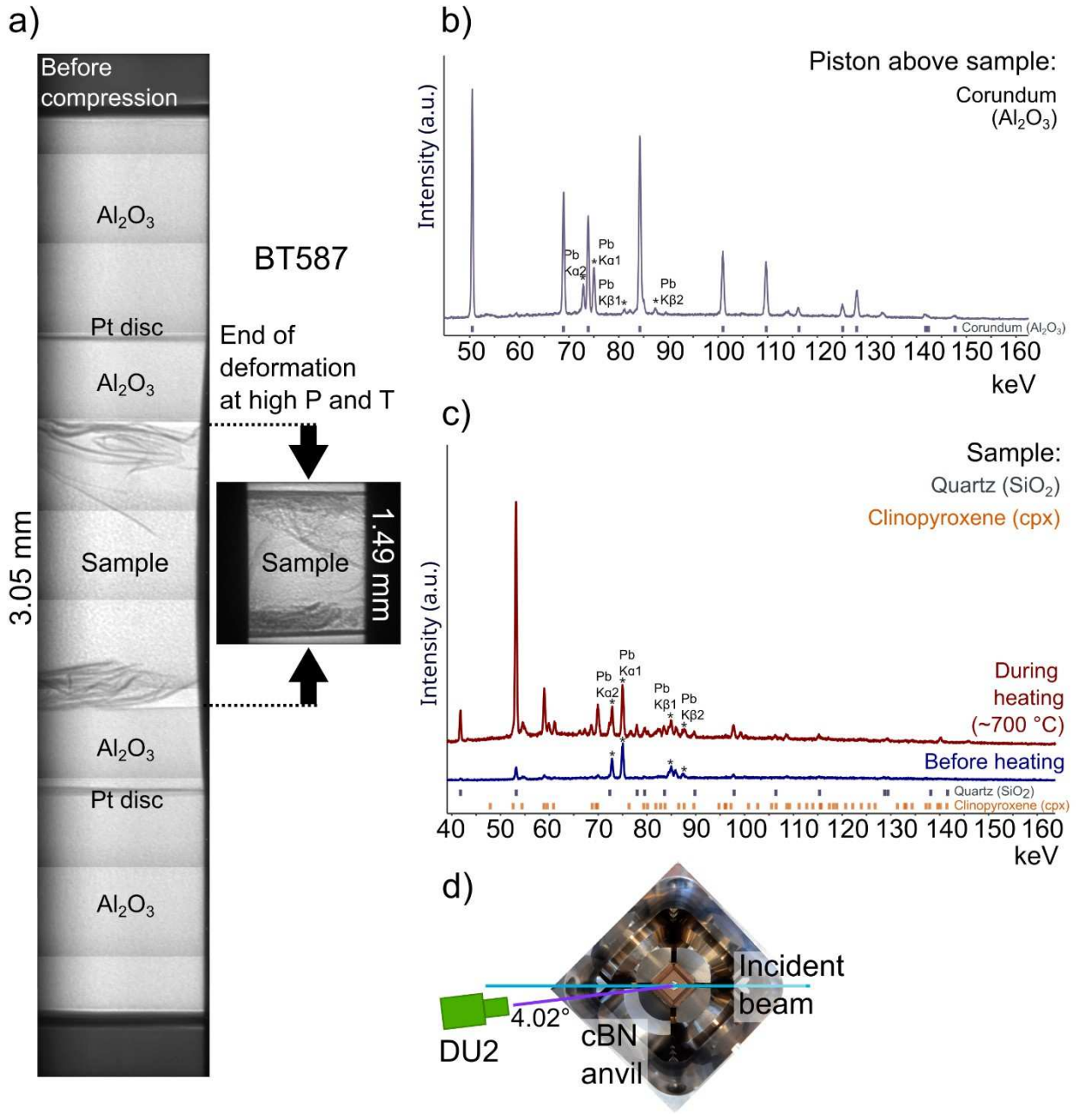
558

559 **Figure 6.** Typical trigger-based waveforms of six sensors over the entire triggering window (left). On the
560 middle and right side, the magnified portions of the waveforms are shown with indication for the first P-
561 wave arrival time (solid green vertical line) and first amplitude (dash red vertical line).



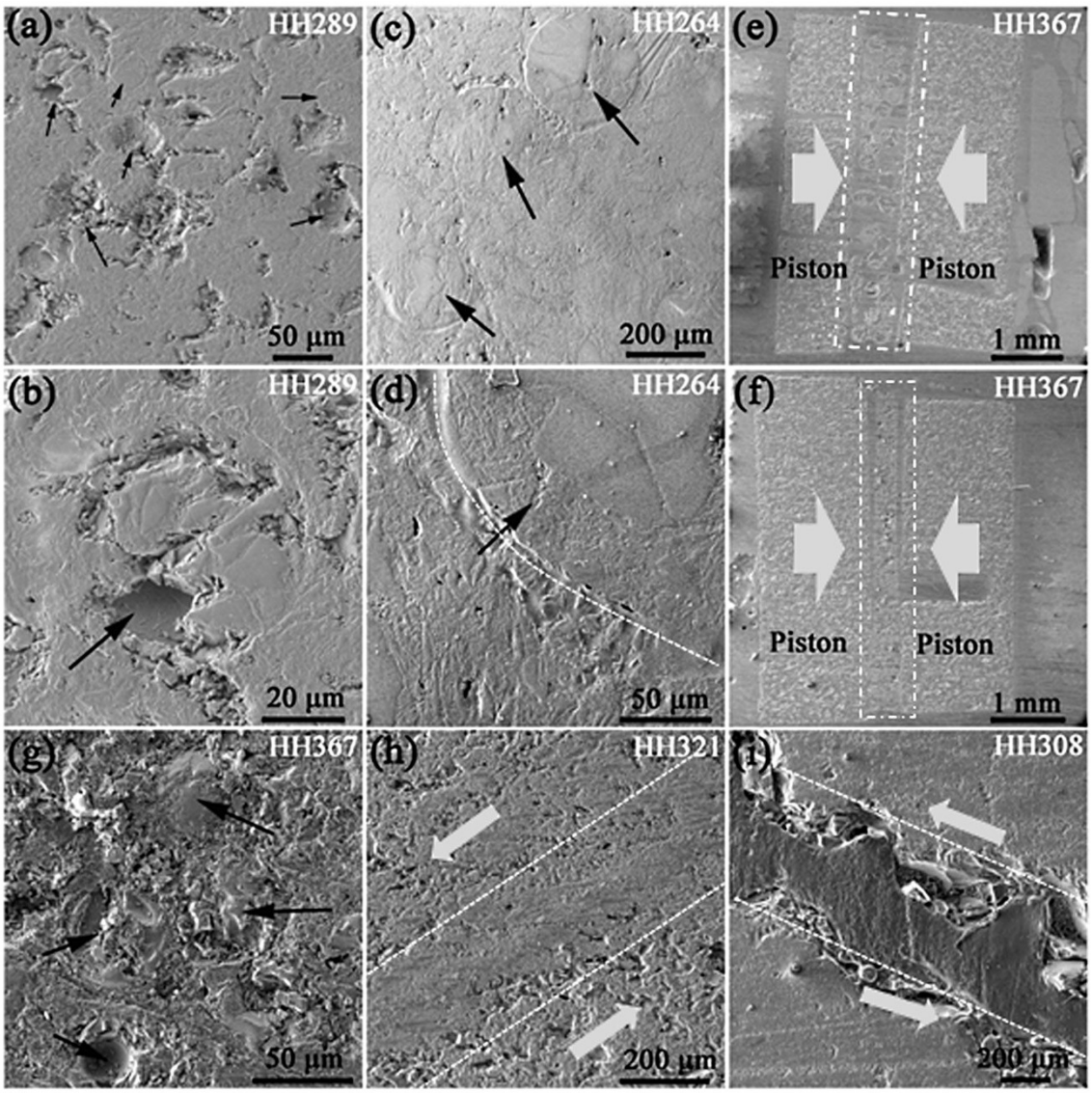
562

563 **Figure 7.** The AE hypocenter locations of (a) large glass beads (b) small glass beads (c) fused glass wedge
564 (d) fused glass wedge + glass beads (e) alumina wedge (f) two layers of glass beads (g) molybdenum
565 wedge + glass beads (h) large glass beads with beam. The gray area indicates the position and size of the
566 sample. The size of the circles denotes the MARSE energy of events. “n” is the number of reprocessed
567 events. The optical images of the self-designed AE sources are shown in the insets (if recovered). The
568 amplitude distribution of each channel is shown at the right side with different colors.



569

570 **Figure 8.** (a) X-ray radiography images of the whole assembly before compression (left) and of the
571 sample after deformation (right). The energy dispersive diffraction patterns of (b) corundum and (c) glass
572 beads sample with some other impurity phases (e.g. clinopyroxene) present. (d) The configuration of
573 incident beam, c-BN anvil and detector during the experiment of BT587 with X-ray beam. The labels 'Pb'
574 in the X-ray diffraction patterns denote the fluorescence peaks from the lead shielding of the detectors.



575

576 **Figure 9**. The SEM images of calibration sources after experiment. (a)(b) large and (c)(d) small glass beads
577 (e-g) two layers of glass beads (h) dense alumina piston (i) fused glass wedge. The numbers of
578 experiments are shown in the corner.

579

Table 1. Experimental runs and conditions.

Run #	Load (MN)	Sample	Active sensors	PTM	AE hits*	AE events*	Strain (%)	Anvil disp. rate (µm/min)$
HH259	0.7	Glass beads (< 100 µm)	6	Pyrophyllite	yes/noise	none	30	36.7
HH264	1.1	Glass beads (> 200 µm)	6	Pyrophyllite	yes/noise	none	30	18.3
HH287	1.3	Glass beads (> 200 µm)	6	B-epoxy	> 60,000 >	4221	15	10
HH289	1.7	Glass beads (< 100 µm)	6	B-epoxy	130,000	6800	15	10
HH290	1.7	Glass beads (> 200 µm)	5	B-epoxy	yes/noise	none	---	---
HH292	1.1	Glass beads (< 100 µm)	5	B-epoxy	> 7,000	40	---	---
HH294	1.3	Fused glass wedge (FGW)	6	B-epoxy	> 54,000	5118|1892	shear	10
HH295	1.3	FGW + hBN	6	B-epoxy	493	2|3	shear	10
HH308	1.3	FGW + glass beads (< 100 µm)	5	B-epoxy	> 24,000	3691	shear	10
HH321	1.3	Dense Al_2O_3 wedge	5	B-epoxy	> 7,000	441	shear	10
HH367	1.3	Two layers glass beads	6	B-epoxy	> 47,000	5655|133	n/a	10
HH411	0.7	Mo piston + glass beads	6	B-epoxy	>1,000	88|71	shear	10
HH413	1.3	---	6	B-epoxy	341	none	6	10
HH420	1.3	---	6	Pyrophyllite	616	none	6	10
BT540	1.2	Glass beads (> 200 µm)	6	B-epoxy	> 94,000	5351	20	10
BT587†	1.1	Glass beads (mixture)	4†	B-epoxy	> 14,000	> 385	50	9

*If two numbers are shown, then AEs for compression and deformation applies. Elsewise, only for compression.

†*In situ* XRD AE experiment with 1 cBN anvil (no sensor). Also, 1 sensor failed on a WC anvil.

$General duration is 45-60 min.

580

581

Acknowledgements The authors gratefully thank discussions and support from Dr. Bhat at P61B and an internal reviewer at DESY. We acknowledge DESY (Hamburg, Germany), a member of the Helmholtz Association HGF, for the provision of experimental facilities. This research was carried out at beamline P61B (Proposal No. I-20210227 for beam time) with the support from the Federal Ministry of Education and Research, Germany (BMBF, grants no.: 05K16WC2 & 05K13WC2). We would like to thank J. Arno and R. S. Snata for SEM sample preparing and SEM measurements.

Declarations The authors have no conflicts to disclose. S.M. and R.F. designed and carried out the experiments. S.M. and R.F. co-wrote the manuscript with contributions from other co-authors.

References

[1] R. Farla, S. Bhat, S. Sonntag, A. Chanyshev, S. Ma, T. Ishii, Z. Liu, A. Néri, N. Nishiyama, G.A. Faria, T. Wroblewski, H. Schulte-Schrepping, W. Drube, O. Seeck, and T. Katsura, J Synchrotron Rad **29**, 409 (2022).
[2] Q. Huang, D. Yu, B. Xu, W. Hu, Y. Ma, Y. Wang, Z. Zhao, B. Wen, J. He, Z. Liu, and Y. Tian, Nature **510**, 250 (2014).
[3] J. Gasc, N. Hilairet, T. Yu, T. Ferrand, A. Schubnel, and Y. Wang, Earth and Planetary Science Letters **474**, 138 (2017).
[4] G. Shen and H.K. Mao, Rep. Prog. Phys. **80**, 016101 (2017).
[5] Z. Jing, T. Yu, M. Xu, J. Chantel, and Y. Wang, Minerals **10**, 126 (2020).
[6] G. Manthilake, N. Bolfan-Casanova, D. Novella, M. Mookherjee, and D. Andrault, Science Advances **2**, e1501631 (2016).
[7] N. Hilairet, Y. Wang, T. Sanehira, S. Merkel, and S. Mei, Journal of Geophysical Research: Solid Earth **117**, (2012).
[8] D.F. Bahr and W.W. Gerberich, J. Mater. Res. **13**, 1065 (1998).
[9] S. Barr and M.L. Benzeggagh, 8 (n.d.).
[10] M.N. Bassim, S.St. Lawrence, and C.D. Liu, Engineering Fracture Mechanics **47**, 207 (1994).
[11] M. Drissi-Habti and M. Gomina, Journal of Alloys and Compounds **188**, 259 (1992).
[12] C. Godano, E. Lippiello, and L. de Arcangelis, Geophysical Journal International **199**, 1765 (2014).
[13] J.A. Hudson, R.G. Pearce, and R.M. Rogers, J. Geophys. Res. **94**, 765 (1989).
[14] T. Ohuchi, X. Lei, H. Ohfuji, Y. Higo, Y. Tange, T. Sakai, K. Fujino, and T. Irifune, Nature Geosci **10**, 771 (2017).
[15] F. Waldhauser, Bulletin of the Seismological Society of America **90**, 1353 (2000).
[16] A. Schubnel, F. Brunet, N. Hilairet, J. Gasc, Y. Wang, and H.W. Green, Science **341**, 1377 (2013).
[17] Y. Wang, L. Zhu, F. Shi, A. Schubnel, N. Hilairet, T. Yu, M. Rivers, J. Gasc, A. Addad, D. Deldicque, Z. Li, and F. Brunet, Sci. Adv. **3**, e1601896 (2017).
[18] T. Officer and R.A. Secco, Physics of the Earth and Planetary Interiors **300**, 106429 (2020).
[19] T.P. Ferrand, N. Hilairet, S. Incel, D. Deldicque, L. Labrousse, J. Gasc, J. Renner, Y. Wang, H.W. Green II, and A. Schubnel, Nat Commun **8**, 15247 (2017).
[20] C. Meade and R. Jeanloz, Nature **339**, 616 (1989).
[21] C. Meade and R. Jeanloz, Science **252**, 68 (1991).

[22] J. Gasc, A. Schubnel, F. Brunet, S. Guillon, H.-J. Mueller, and C. Lathe, Physics of the Earth and Planetary Interiors **189**, 121 (2011).
[23] D.P. Dobson, P.G. Meredith, and S.A. Boon, Science **298**, 1407 (2002).
[24] H. Jung, Y. Fei, P.G. Silver, and H.W. Green, Review of Scientific Instruments **77**, 014501 (2006).
[25] T. Ohuchi, X. Lei, H. Ohfuji, Y. Higo, Y. Tange, T. Sakai, K. Fujino, and T. Irifune, Nature Geosci **10**, 771 (2017).
[26] A.A. De Ronde, D.P. Dobson, P.G. Meredith, and S.A. Boon, Geophysical Journal International **171**, 1282 (2007).
[27] M. Shigeishi and M. Ohtsu, Construction and Building Materials **15**, 311 (2001).
[28] M. Ohtsu, Research in Nondestructive Evaluation **6**, 169 (1995).
[29] M. Ohtsu, J. Geophys. Res. **96**, 6211 (1991).
[30] A.K. Singh, C. Balasingh, H. Mao, R.J. Hemley, and J. Shu, Journal of Applied Physics **83**, 7567 (1998).
[31] S. Ono, High Pressure Research **38**, 414 (2018).
[32] Q. Li, *MIT Rock Mechanics Group AE Code* (n.d.).
[33] T. Officer and R.A. Secco, Physics of the Earth and Planetary Interiors **300**, 106429 (2020).

MIT Rock Mechanics Group AE Code (https://www.mathworks.com/matlabcentral/fileexchange/72339-mit-rock-mechanics-group-ae-code), MATLAB Central File Exchange. Retrieved April 5, 2022.

Supporting information

Acoustic emission detection of micro-cracks under high pressure and high temperature in a deformation large-volume apparatus

Shuailing Ma[1,2], Julien Gasc[3], Sarah Incel[4], Stefan Sonntag[1], Robert Farla[1]

1. Deutsches Elektronen-Synchrotron DESY, Notkestr, 85, 22607, Hamburg, Germany

2. Institute of High Pressure Physics, School of Physical Scientific and Technology, Ningbo University, Ningbo, 315211, China

3. Synergetic Extreme Condition High-Pressure Science Center , State Key Laboratory of Superhard Materials, college of physics, Jilin University, Changchun 130012, China

4. Laboratoire de Géologie, CNRS – École Normale Supérieure, 24 rue Lhomond, 75005 Paris, France

5. Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany

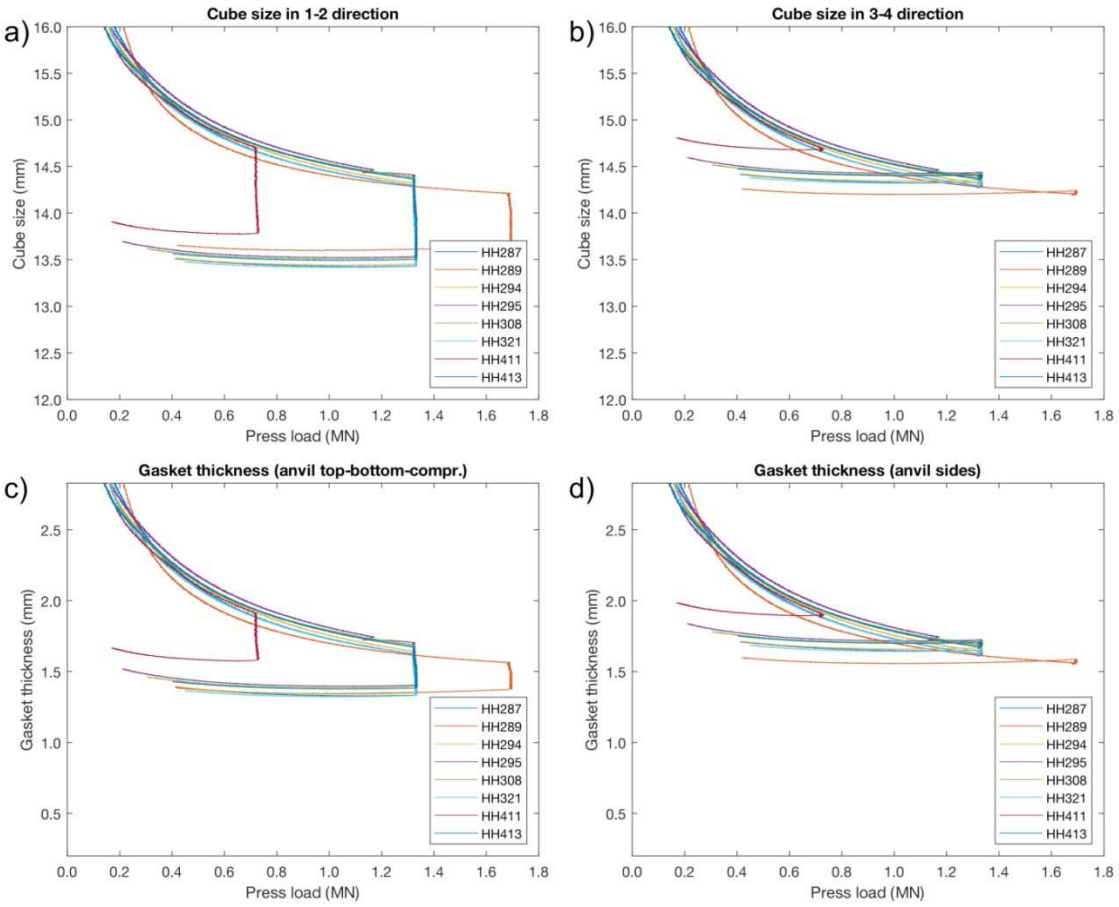


Figure S1. Calculation of the change of cube size and gasket thickness using the logged press data after each experiment. (a) Cube size in the No1-No2 direction (vertical direction), (b) Cube size in the No3-No4 direction (horizontal direction), (c) gasket thickness of anvil top-bottom-compression and (d) gasket thickness of side anvils with the function of oil pressure at different number experiment.

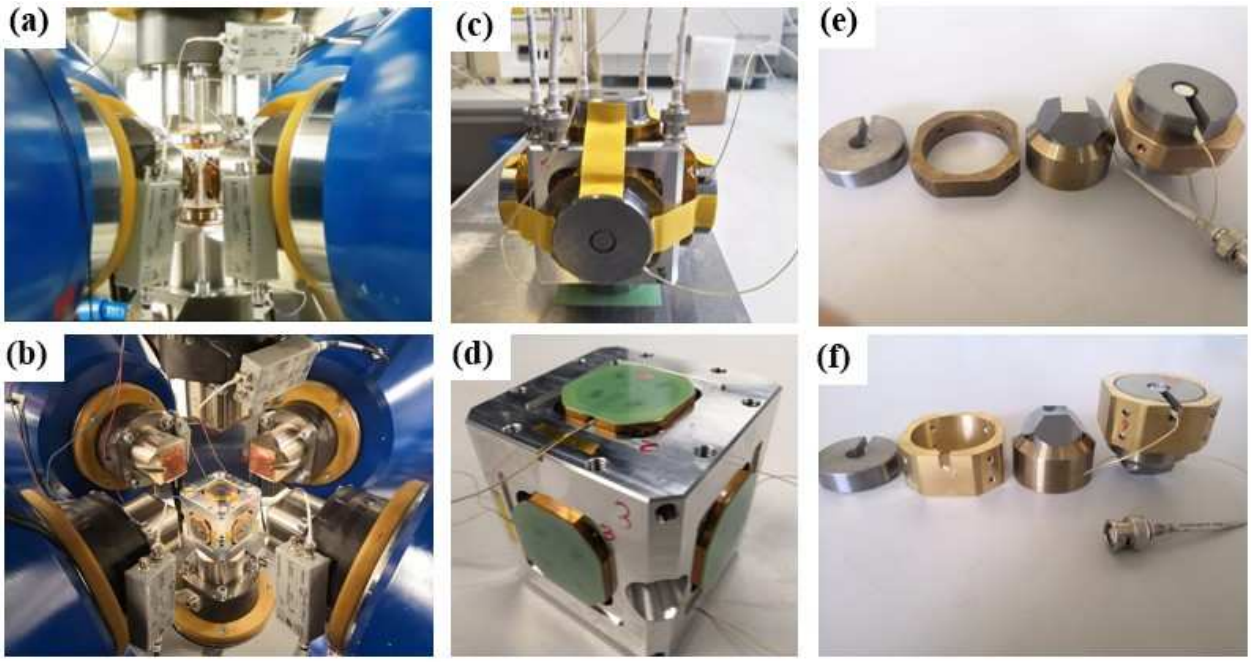


Figure S2. Two different types of alignment frames used during the development of the AE system. (a) (b) Two different type of AEs systems in the press. (c) (d) the comparison of two different type of alignment frames. Two different length of the brass alignment rings to hold the tungsten carbide anvils and supporting rings. (e) Lower and (f) higher brass alignment rings.

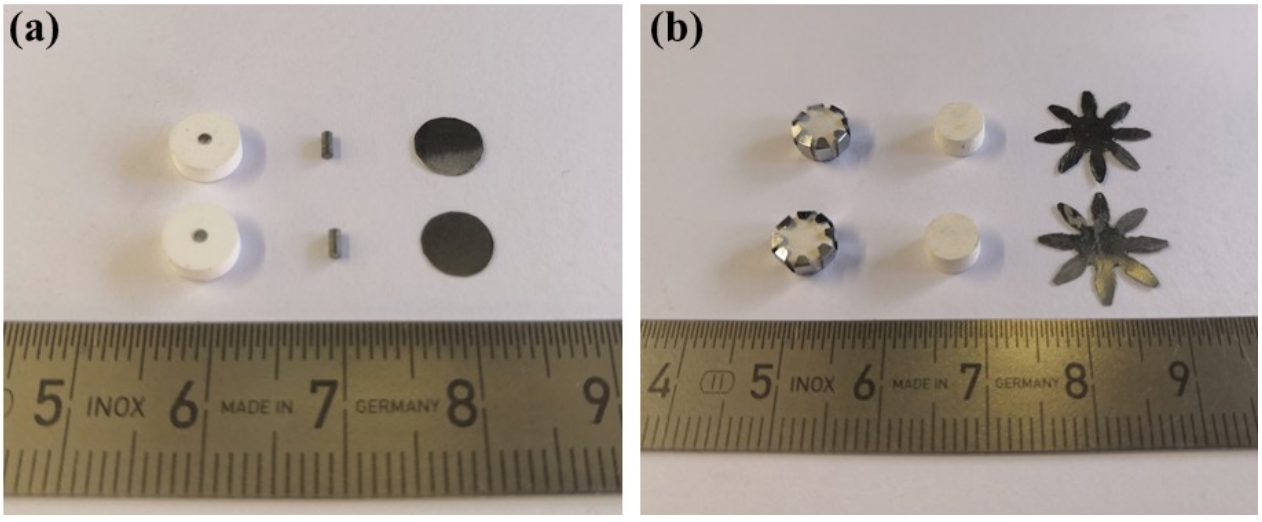


Figure S3. The different types of electrode for acoustic emission assembly. (a) zirconia plug, molybdenum electrode, molybdenum foil from left to right. (b) The wrapped zirconia plug, zirconia plug, star-shaped molybdenum foil.

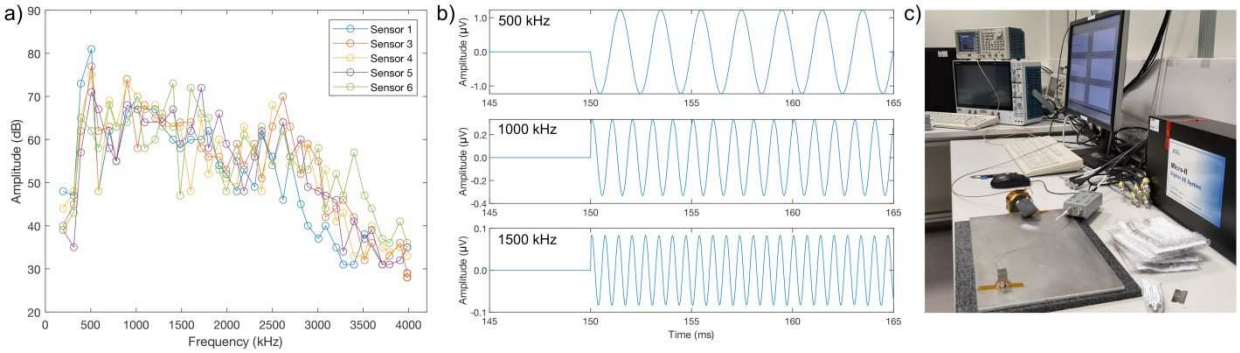


Figure S4. (a) The amplitude of six sensors with the function of frequency (b) The waveform of trigger source (c) the image of sensitivity test experiment.

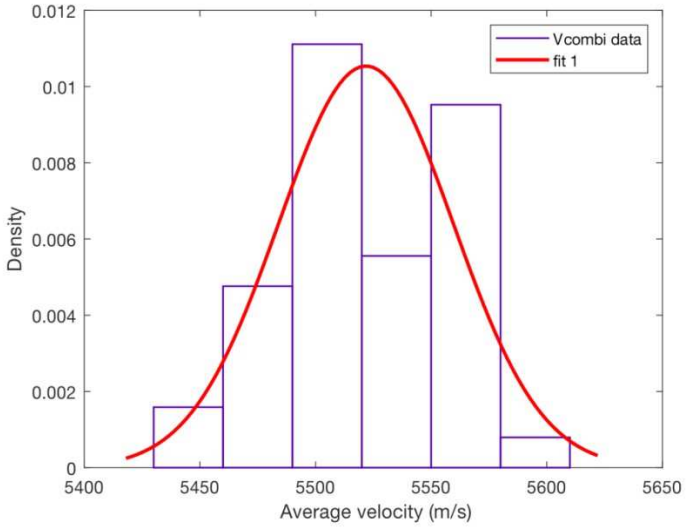
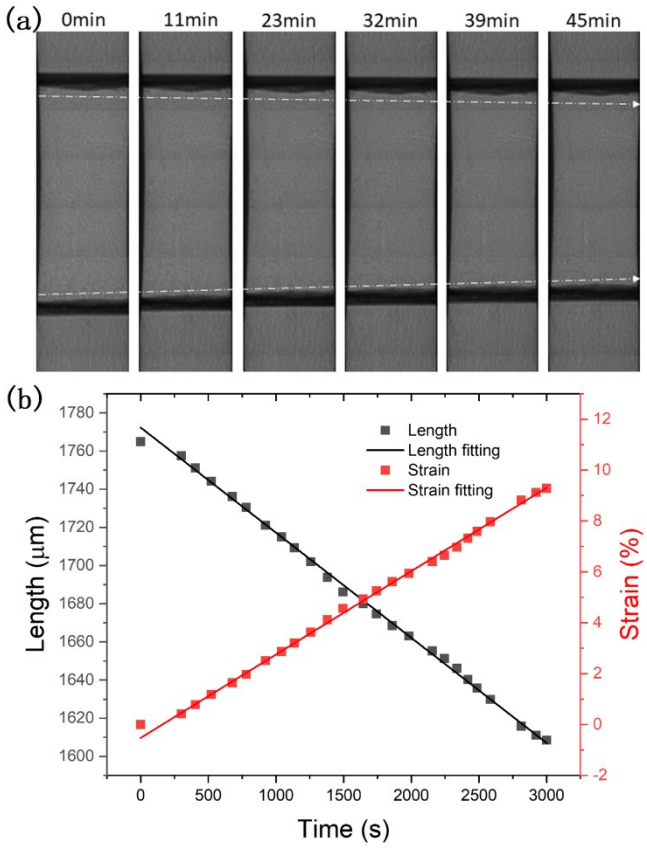


Figure S5. The density distribution of average velocity of boron epoxy pressure medium.



689

690 **Figure S6.** (a) The X-ray radiography image of the glass beads sample of BT540 at different deformation
691 time. (b) The length of the sample that deduced from the X-ray radiography image and the strain history
692 with time.

693

694

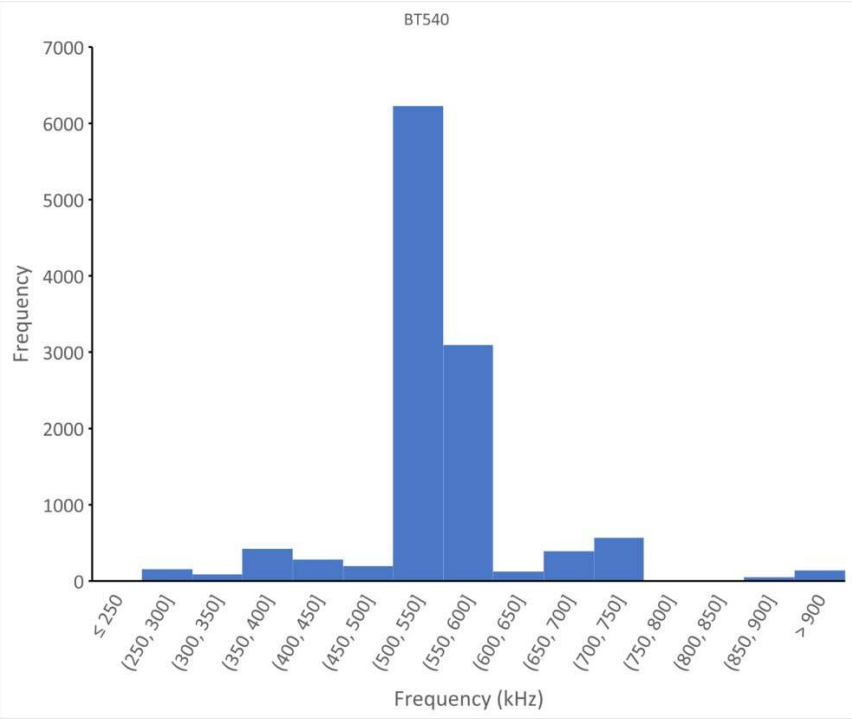


Figure S7. The histogram of frequency versus the acoustic emission signal frequency of BT540.

Table S1. Data collection and processing criteria.

AEwin filters	
Preamp gain	40 dB
1st threshold crossing (FTC)	30 dB ± 6 FTBND
Sampling rate	40 MS/s (i.e. 25 ns)
Data length	15k
Timing parameters	
Peak definition time (PDT)	50 µs
Hit definition time (HDT)	200 µs
Hit lockout time (HLT)	300 µs
Maximum duration	100 ms
Analog filters	
Frequency range	0.2 to 3 MHz
Energy range	2 to 99999
Av. Frequency range	101 to 999 kHz
Data extraction filters	
Counts	> 100
Rise time	< 60 µs
Duration*	60 - 1200 s until end
Amplitude	< 95 (100 = saturation)
Additional MATLAB filters	
Minimum # of sensors	5
Av. Velocity	5520 m/s
Sensor to source distance†	44 mm, 44.5 mm
Data interpolation (from 25 ns)	2 ns (i.e. 500 MS/s)
New threshold crossing	0.002 v (i.e. 26 dB)

*Due to initial cracking of assembly and gaskets, the first few minutes are removed before data processing in MATLAB.

†Depending on cube size under load. Anvil length is 37 mm.

Supplementary information

MATLAB script used to re-process the 3d location of AEs.

```
%% 3d locations of Acoustic Emissions recorded in the Aster-15 LVP at P61B.
%
% Author 1: R. Farla (robert.farla@desy.de)
% Author 2: J. Gasc (gasc@biotite.ens.fr)
% Date: Autumn 2021
% Version 1.1
%
% Description: This script will read all exported event waveforms from
% AEwin, obtained after beam time. It resamples the time interval from
% 40 MS/s (25 ns) to 500 MS/s (2 ns), then detects the 1st crossing of the
% threshold (user defined) and proceeds to plot all the events in a virtual
% cylinder representing the sample (user defined) in a 3d plot.
```

```
% It will report the number/percentage of events whose locations were
% successfully determined by regression / minimisation of the function as
% given in the AEwin manual.
%
% References: The script uses code from the MIT Rock Mechanics Group:
% Qiuyi Li (2021). MIT Rock Mechanics Group AE Code
% https://www.mathworks.com/matlabcentral/fileexchange/72339-mit-rock-mechanics-group-ae-code)
% MATLAB Central File Exchange.
%
% Input data filename format: name_sensor_event_timestamp.csv
% 1. name = run number of experiment (NO dash '-' or underscore '_' ONLY HHxxx or BTxxx or HHxxxcompression,
etc.)
% 2. sensor = sensor number (1 to 6)
% 3. event = event number
% 4. timestamp
% file format = csv, exported with no index column.
%   Note: the pre-trigger time can be determined (13th row) by exporting a
%   waveform with 'Time of sample relative to trigger (T=0)' option.
%   NB: TIME OF TEST is the time where the voltage signal crossed the
%   threshold for the first time in each AE recording.
%
% DO NOT IMPORT TENS OF THOUSANDS OF FILES!! FILTER THE OUTLIERS IN AEWIN FIRST!
% A MAX. ACCEPTABLE AMOUNT IS 1000 EVENTS (6000 FILES/HITS).
% -> What to filter? See examples below:
% 1. Counts filter: < 100 (and exclude outliers)
% 2. Rise-time filter: About 1 to 80 us (avoid unrealistically long AEs).
% 3. Time filter: For compression AE data, enable this filter e.g. > 600 s
% 4. Duration filter (less effective): e.g. > 2000 s (not too long AEs)
%
% Parameters: The user may adjust the following parameters below.
%
%% User section
% Enter default directory for easy file access
wpath = 'F:\DESY Beamline\Laboratory\Experiments\Acoustic Emissions studies\Rawdata\';

%--Set threshold for first crossing (in volts)--%
% If too many events are not plotted, the threshold could be too low
% and you are triggering the noise.
th = 0.002; %v
% This corresponds to an amplitude (dB) of:
% ampl = 20 * log10(th/1.e-6) - 40 %where 40 dB is the default preamp gain setting.
% AEwin is configured to trigger at 30 dB (FTC) by default.

%--Set pre-trigger time--%
pretriggertime = 150.e-6; %s, where 150 us is the pre-trigger time by default.
%pretriggertime = 256.e-6;

%--Set average wavespeed--%
% Before AE recording, perfom a pulse test (APMG) to obtain the average
% wavespeed in the assembly (in all 3 directions). Enter the number below.
v = 5520; %m/s

%--Set sensor position to cube centre--%
% Knowing the precise cube size under target press load is helpful, e.g. using X-ray radiography.
sensordist = 44.0; %mm

%%--Set minimum number of sensors used in 3d location calculation (4 to 6)--%
sensormin = 6;

%%--Set distance error for 3d location calculation--%
disterror = 10; %mm (Too small value will result in no events plotted)

%--Set cylinder size (representing sample) in 3d plot--%
radius = 2.0; %mm
height = 4.0; %mm (assuming the cylindical sample is exactly in the middle of the assembly)

%%-Set plot dimensions in x, y, z (1:1:1 aspect ratio) both +ve and -ve directions--%
u = 6.0; %mm

%--Enable/Disable normalization of hit characteristics (for more distinct range of marker size in 3d plots)--%
NORM = 1; % 1 = enabled, 0 = disabled

%--Enable/Disable filter for saturated events (amp = 98 to 100 dB), applies to all plots--%
FT = 1; % 1 = enabled, 0 = disabled

%--Set marker size multiplication factor--%
MF = 30; %Might have to tweak this value for best results per dataset and conditions above

%--3d plot orientation--%
% Enable any for additional viewing directions of the 3d location plot.
Xview = 0; % 1 = enabled, 0 = disabled
```

```
Yview = 0; % 1 = enabled, 0 = disabled
Zview = 0; % 1 = enabled, 0 = disabled

%--Moment Tensor Inversion calculations--%
MT = 0; %1 = enabled, 0 = disabled
density = 2900; %g/cc

%% \/ Code section (no need to modify) \/ %%
%% import of csv waveform files %%
%--Browse to directory of the waveform files--%
uiwait(msgbox("Select directory with waveforms","Information","help"));
input_dir = uigetdir(wpath,"Select directory with waveforms");
%--Browse to directory of AE hit parameters file--%
uiwait(msgbox("Select AE parameter file","Information","help"));
[filename,path] = uigetfile('*.txt','Select a file',wpath);
% match = [".txt",".TXT"];
% filenm = erase(filename,match);
if contains(filename,"HH") == 1
    pat = "HH" + digitsPattern(3);
    filenm = string(extract(filename,pat));
elseif contains(filename,"BT") == 1
    pat = "BT" + digitsPattern(3);
    filenm = string(extract(filename,pat));
else
    return
end

all_files=dir([input_dir,'\*.csv']);
nb_pzt=6;
nb_event = length(all_files)./nb_pzt;
nevt = nb_event(:,end);
nb_points=15360;

%% import of AE hit parameter file %%
startRow = 7;
formatSpec = '%3f%21f%4f%6f%6f%6f%10f%4f%6f%4f%12f%f%[^\n\r]';
fileID = fopen(append(path,filename),'r');
textscan(fileID, '%[^\n\r]', startRow-1, 'WhiteSpace', '', 'ReturnOnError', false, 'EndOfLine', '\r\n');
dataArray = textscan(fileID, formatSpec, 'Delimiter', '', 'WhiteSpace', '', 'TextType', 'string', 'EmptyValue',
NaN, 'ReturnOnError', false);
AEchars = [dataArray{1:end-1}]; %Sorted by TIME OF TEST
clearvars filename startRow formatSpec fileID dataArray ans;
% ID(col1) SSSSSSSS.mmmuuun(col2) CH(col3) RISE(col4) COUN(col5) ENER(col6)
% DURATION(col7) AMP(col8) A-FRQ(col9) THR(col10) ABS-ENERGY (col11)
% P-FRQ(col12)
ind1 = AEchars(:,3) == 1;
ind2 = AEchars(:,3) == 2;
ind3 = AEchars(:,3) == 3;
ind4 = AEchars(:,3) == 4;
ind5 = AEchars(:,3) == 5;
ind6 = AEchars(:,3) == 6;

Ch1 = AEchars(ind1,:);
Ch2 = AEchars(ind2,:);
Ch3 = AEchars(ind3,:);
Ch4 = AEchars(ind4,:);
Ch5 = AEchars(ind5,:);
Ch6 = AEchars(ind6,:);

Chave = (Ch1 + Ch2 + Ch3 + Ch4 + Ch5 + Ch6) / 6;

%% Normalize data for 3d plots
if NORM == 1
    cnt_norm = normalize(Chave(:,5),'range') .* MF;
    en_norm = normalize(Chave(:,6),'range') .* MF;
    dur_norm = normalize(Chave(:,7),'range') .* MF;
    amp_norm = normalize(Chave(:,8),'range') .* MF;
elseif NORM == 0
    cnt_norm = Chave(:,5)./ MF;
    en_norm = Chave(:,6)./ MF;
    dur_norm = Chave(:,7)./ MF;
    amp_norm = Chave(:,8);
end

%Combine all normalized event data into a new matrix
Chave_norm = [amp_norm,en_norm,dur_norm,cnt_norm];
Chave_norm(Chave_norm == 0) = NaN;

% Scale normalized data to show desired marker size in 3d plots
amplitude = Chave_norm(:,1);
energy = Chave_norm(:,2);
```

```
duration = Chave_norm(:,3);
counts = Chave_norm(:,4);

%% Loop files to extract AE and Channel number
Out_Mat=NaN(nb_event,nb_points,nb_pzt);
for mnop=1:length(all_files)
    filename=[input_dir,'\',all_files(mnop).name];

    str = all_files(mnop).name;
    exp_pzt = '(?<!\d)(\d)(?!\d)';
    tmp_pzt = str2double(regexp(str,exp_pzt,'match','once'));
    exp_evt = '(?<=_)\d+(?=_)';
    tmp_evt = str2double(regexp(str,exp_evt,'match'));
    tmp_event = tmp_evt(:,2);

    fileID = fopen(filename);
    [data] = cell2mat(textscan(fileID, '%.8f','emptyvalue',NaN,'headerlines',11));
    fclose(fileID);

    Out_Mat(tmp_event,1:length(data),tmp_pzt)=data;
end

%sampling time in seconds
tstep=1/40000000;
%time scale
time=(1:nb_points)*tstep;

%% Plot the first AE's first channel
n = 1;
figure
plot(time,Out_Mat(n,:,1))
% hold on
% plot(time,Out_Mat(n,:,2))
% hold on
% plot(time,Out_Mat(n,:,3))
% hold on
% plot(time,Out_Mat(n,:,4))
% hold on
% plot(time,Out_Mat(n,:,5))
% hold on
% plot(time,Out_Mat(n,:,6))
ylim([-0.1 0.1]) %mm

%% Loop files to extract time stamp
time_stamps=NaN(nb_event,nb_pzt);
time_stamps2=NaN(nb_event,nb_pzt);
for mnop=1:length(all_files)
    filename=[input_dir,'\',all_files(mnop).name];

    str = all_files(mnop).name;
    exp_pzt = '(?<!\d)(\d)(?!\d)';
    tmp_pzt = str2double(regexp(str,exp_pzt,'match','once'));
    exp_evt = '(?<=_)\d+(?=_)';
    tmp_evt = str2double(regexp(str,exp_evt,'match'));
    tmp_event = tmp_evt(:,2);

    myline=11;
    fid = fopen(filename);
    if fid ~= -1
        for i=1:(myline-1)
            fgetl(fid);
        end
        timeline=fgetl(fid);
        fclose(fid);
    end
    format long
    time_stamps(tmp_event,tmp_pzt)=str2double(timeline(end-11:end));
    time_stamps2(tmp_event,tmp_pzt)=str2double(timeline(end-9:end));
    x = isnan(time_stamps);
    time_stamps(x) = time_stamps2(x);
end
% Need to subtract the pre-trigger time from the time stamps in order to
% get the time at the first voltage value (row 1)
time_stamps = time_stamps - pretriggertime; % where 150 us is the pre-trigger time by default

%% Restrict in t range
tmin=find(time>=1.3e-4,1);
tmax=find(time>=1.7e-4,1);
```

```
%tmin=find(time>=1.e-4,1);
%tmax=find(time>=3.e-4,1);
t2=time(tmin:tmax);

trunc=zeros(nb_event,length(t2),nb_pzt);
for i=1:nb_event(:,end)
    for j=1:nb_pzt
    trunc(i,:,j)=Out_Mat(i,tmin:tmax,j);
    end
end

%% Interpolation (re-sampling) for more accurate dt determination
interp_tstep=2.e-9; %resampling with 2 ns instead of original 25 ns

tinterp=time(tmin):interp_tstep:time(tmax);

trunc_interp=zeros(nb_event,length(tinterp),nb_pzt);
for i=1:nb_event(:,end)
    for j=1:nb_pzt
        trunc_interp(i,:,j)=interp1(t2,trunc(i,:,j),tinterp,'spline');
    end
end

%% Visualize first 30 AEs truncated and their interpolated signal
% figure
% for j=14:14
% for i=1:nb_pzt
%     subplot(6,1,i)
%     %plot(t2,trunc(j,:,i));
% hold on;
% plot(tinterp,trunc_interp(j,:,i));
% ylim([-0.005 0.005]) %mm
% xlim([1.495e-4 1.505e-4])
% end
% %pause
% %clf
% end

%% Find first arrivals for all sensors (all events) - FTC method
th2 = -th;
%Loop for all sensors
solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_1 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,1) >= th);
    idx2 = find(trunc_interp(I,:,1) <= th2);
    solution(I,:) = idx(1);
    solution2(I,:) = idx2(1);
    MinFTC_1(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_2 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,2) >= th);
    idx2 = find(trunc_interp(I,:,2) <= th2);
    solution(I,:) = idx(1);
    solution2(I,:) = idx2(1);
    MinFTC_2(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_3 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,3) >= th);
    idx2 = find(trunc_interp(I,:,3) <= th2);
    solution(I,:) = idx(1);
    solution2(I,:) = idx2(1);
    MinFTC_3(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_4 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,4) >= th);
    idx2 = find(trunc_interp(I,:,4) <= th2);
    solution(I,:) = idx(1);
```

```
    solution2(I,:) = idx2(1);
    MinFTC_4(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_5 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,5) >= th);
    idx2 = find(trunc_interp(I,:,5) <= th2);
    solution(I,:) = idx(1);
    solution2(I,:) = idx2(1);
    MinFTC_5(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

solution = zeros(nb_event(:,end),1);
solution2 = zeros(nb_event(:,end),1);
MinFTC_6 = zeros(nb_event(:,end),1);
for I = 1:nb_event(:,end) % I = number of events
    idx = find(trunc_interp(I,:,6) >= th);
    idx2 = find(trunc_interp(I,:,6) <= th2);
    solution(I,:) = idx(1);
    solution2(I,:) = idx2(1);
    MinFTC_6(I,:) = (min(solution(I,:), solution2(I,:)) * interp_tstep) + 1.3e-4;
end

MinFTC = [MinFTC_1, MinFTC_2, MinFTC_3, MinFTC_4, MinFTC_5, MinFTC_6];
arrivalArray = time_stamps + MinFTC;

%% Moment Tensor Inversion (for HybridMT code)
if MT == 1
    % Find area under the 1st peak of all hits from all sensors
    % For positive 1st peak
    omega_pos = zeros(nb_event,6);
    omegaloc_pos = zeros(nb_event,6);
    for n = 1:nb_event(:,end)
        for m = 1:nb_pzt
        [pks,locs,w,p] = findpeaks(trunc_interp(n,:,m),tinterp,'MinPeakHeight',0.005,'NPeaks',1);
        omegaloc_pos(n,m,:) = locs;
        omega_pos(n,m,:) = w * pks;
        end
    end
    % For negative 1st peak
    omega_neg = zeros(nb_event,6);
    omegaloc_neg = zeros(nb_event,6);
    for n = 1:nb_event(:,end)
        for m = 1:nb_pzt
        [pks,locs,w,p] = findpeaks(-trunc_interp(n,:,m),tinterp,'MinPeakHeight',0.005,'NPeaks',1);
        omegaloc_neg(n,m,:) = locs;
        omega_neg(n,m,:) = -(w * pks);
        end
    end
    % Process Omega values to identify the correct first peak (+ve or -ve)
    MinOmega = min(omegaloc_pos,omegaloc_neg); %find which peak comes first
    MinOmega(MinOmega ~= omegaloc_pos) = NaN; %Set all locations for the negative peaks to NaN
    Omgidx1 = ~isnan(MinOmega); % value is not NaN
    MinOmega(Omgidx1) = omega_pos(Omgidx1); %replace all location values with Omega values for +ve peaks
    Omgidx2 = isnan(MinOmega); % value is NaN
    MinOmega(Omgidx2) = omega_neg(Omgidx2); %replace all NaN values with Omega values for -ve peaks
    % MinOmega is, hence, the matrix for the area (Omega) under the first peaks,
    % correctly identified with +ve or -ve sign.
else
% do nothing
end
%% MIT group 3d location code
initXYZ = [0.00, 0.00, 0.00]; %m
distError = [disterror * 1.e-3, disterror * 1.e-3, disterror * 1.e-3, disterror * 1.e-3, disterror * 1.e-3,
disterror * 1.e-3]; %m
sensor = sensordist * 1.e-3;
sensors = [0 0 sensor; 0 0 -sensor; sensor 0 0; -sensor 0 0;  0 sensor 0; 0 -sensor 0;]; %m
[x,y,z,~,e] = tradLoc3D(arrivalArray,v,sensors,distError,sensormin,6,initXYZ);

%% Processing with additional data
if MT == 1
    comb_energy = [x y z energy MinOmega];
    comb_amplitude = [x y z amplitude MinOmega];
    comb_duration = [x y z duration MinOmega];
    comb_counts = [x y z counts MinOmega];
elseif MT == 0
    comb_energy = [x y z energy];
    comb_amplitude = [x y z amplitude];
```

```
        comb_duration = [x y z duration];
        comb_counts = [x y z counts];
    end
    %% Filtering
    % Remove saturated data (amplitude > 98)
    if FT == 1
        ind_amp = find(any(Chave(:,8) > 98,2));
        comb_energy(ind_amp,:) = [];
        comb_amplitude(ind_amp,:) = [];
        comb_duration(ind_amp,:) = [];
        comb_counts(ind_amp,:) = [];
        comb_energy_sum = mean(comb_energy(:,4),'omitnan');
        comb_amplitude_sum = mean(comb_amplitude(:,4),'omitnan');
        comb_duration_sum = mean(comb_duration(:,4),'omitnan');
        comb_counts_sum = mean(comb_counts(:,4),'omitnan');
        comb_energy_norm = comb_energy(:,4) ./ comb_energy_sum;
        comb_amplitude_norm = comb_amplitude(:,4) ./ comb_amplitude_sum;
        comb_duration_norm = comb_duration(:,4) ./ comb_duration_sum;
        comb_counts_norm = comb_counts(:,4) ./ comb_counts_sum;
        comb_energy(:,4) = comb_energy_norm .*MF;
        comb_amplitude(:,4) = comb_amplitude_norm .*MF;
        comb_duration(:,4) = comb_duration_norm .*MF;
        comb_counts(:,4) = comb_counts_norm .*MF;
    else
    % do nothing
    end
    % Remove all unsuccessful data (cannot be located, i.e. zero x,y,z positions)
    out_energy = comb_energy(all(comb_energy,2),:);
    out_amplitude = comb_amplitude(all(comb_amplitude,2),:);
    out_duration = comb_duration(all(comb_duration,2),:);
    out_counts = comb_counts(all(comb_counts,2),:);

    %% Final calculations
    % Cylindrical model of sample
    height = height / 2;
    [Xsample,Ysample,Zsample] = cylinder(radius);
    Zsample(1,:) = -height; %mm
    Zsample(2,:) = height; %mm
    %add optional 3d sliding surface
    % Xp = [-2.8 2.8 2.8 -2.8];
    % Yp = [-2.8 -2.8 2.8 2.8];
    % Zp = [-3 -3 3 3];

    % For output below and later figures
    Arrtime_ave = mean(arrivalArray,2);
    Chave_time = [Arrtime_ave Chave];

    if FT == 1
        indx_amp = find(any(Chave_time(:,9) > 98,2)); % Saturation filter
        Chave_time(indx_amp,:) = [];
    else
    % do nothing
    end

    % Output to command window:
    difper = 100 * (size(out_energy,1) / nb_event(:,end));

    fprintf('<strong>Events plotted: %d </strong>', (size(out_energy,1)))
    fprintf('<strong>of %d </strong>', nb_event(:,end))
    fprintf('(%1.0f', difper)
    fprintf('%%) \n')

    %% Moment Tensor Inversion (for HybridMT code)
    % Write out the input files for hybridMT
    % Each event has a header line, then six lines for each hit on a sensor
    % that make up the event. See the hybridMT documentation for more details.
    if MT == 1
        %azm = 0;
        aoi = 0;
        %tko = 180;
        %calculate distance from AE source to each sensor for all events.
        distsens1 = zeros(size(out_energy,1),1);
        distsens2 = zeros(size(out_energy,1),1);
        distsens3 = zeros(size(out_energy,1),1);
        distsens4 = zeros(size(out_energy,1),1);
        distsens5 = zeros(size(out_energy,1),1);
        distsens6 = zeros(size(out_energy,1),1);
        for i = 1:size(out_energy,1)
            %Sensor 1 (top = 0 0 44.5, postive)
            distsens1(i,:) = sensors(1,3) - out_energy(i,3);
            %Sensor 2 (bottom = 0 0 -44.5, negative)
```

```
            distsens2(i,:) = abs(sensors(2,3) - out_energy(i,3));
            %Sensor 3 (X-dir = 44.5 0 0, positive)
            distsens3(i,:) = sensors(3,1) - out_energy(i,1);
            %Sensor 4 (X-dir = -44.5 0 0, negative)
            distsens4(i,:) = abs(sensors(4,1) - out_energy(i,1));
            %Sensor 5 (Y-dir = 0 44.5 0, positive)
            distsens5(i,:) = sensors(5,2) - out_energy(i,2);
            %Sensor 6 (Y-dir = 0 -44.5 0, negative)
            distsens6(i,:) = abs(sensors(6,2) - out_energy(i,2));
        end
        distsens = [distsens1 distsens2 distsens3 distsens4 distsens5 distsens6];

        % One file (RAW data format)
        mkdir MT
        formatSpec = "%sMT.txt";
        filename = append('MT/', sprintf(formatSpec, filenm));
        fileID = fopen(filename, 'w');
        header = '%s_%04d %.f\r\n';
        lines1 = 'AE1 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        lines2 = 'AE2 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        lines3 = 'AE3 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        lines4 = 'AE4 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        lines5 = 'AE5 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        lines6 = 'AE6 Z P %e %.1f %.1f %.1f %.f %.4f %.f\r\n';
        % lines = '\nAE Z P %e %.6f %.6f %.6f';
        %AEnumber = zeros(size(out_energy,1),1);
        for i = 1:size(out_energy,1)
            fprintf(fileID, header, filenm, i, nb_pzt);
            fprintf(fileID, lines1, out_energy(i,5), 0, aoi, 180, v, distsens(i,1) * 1000, density);
            fprintf(fileID, lines2, out_energy(i,6), 0, aoi, 0, v, distsens(i,2)* 1000, density);
            fprintf(fileID, lines3, out_energy(i,7), 0, aoi, 90, v, distsens(i,3)* 1000, density);
            fprintf(fileID, lines4, out_energy(i,8), 180, aoi, 90, v, distsens(i,4)* 1000, density);
            fprintf(fileID, lines5, out_energy(i,9), 90, aoi, 90, v, distsens(i,5)* 1000, density);
            fprintf(fileID, lines6, out_energy(i,10), 270, aoi, 90, v, distsens(i,6)* 1000, density);
        end
        fclose(fileID);
    else
    % do nothing
    end
    %% Create 3d event location figures
    figure('name','3d location data')
    set(gcf, 'Position',  [100, 100, 1000, 1000])
    t = tiledlayout(2,2);
    t.Padding = 'compact';
    t.TileSpacing = 'compact';

    Min = min(Chave_time(:,7));
    Max = max(Chave_time(:,7));
    ax1 = nexttile;
    data1 = scatter3(ax1,out_energy(:,1)*1e3, out_energy(:,2)*1e3, out_energy(:,3)*1e3, out_energy(:,4), 'filled',
    'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.8500 0.3250 0.0980], 'DisplayName', 'MARSE Energy');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax1,'MARSE Energy')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'ydir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,9));
    Max = max(Chave_time(:,9));
    ax2 = nexttile;
    data1 = scatter3(ax2,out_amplitude(:,1)*1e3, out_amplitude(:,2)*1e3, out_amplitude(:,3)*1e3,
    out_amplitude(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0 0.4470 0.7410], 'DisplayName',
    'Amplitude');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax2,'Amplitude')
    grid on;
```

```
pbaspect([1 1 1])
xlim([-u u]) %mm
ylim([-u u]) %mm
zlim([-u u]) %mm
xlabel('X');
ylabel('Y');
zlabel('Z');
set(gca,'YDir','reverse')
str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
lgnd = legend(data1, 'Location', 'north');
lgnd.FontSize = 8;
hold off;

Min = min(Chave_time(:,8));
Max = max(Chave_time(:,8));
ax3 = nexttile;
data1 = scatter3(ax3,out_duration(:,1)*1e3, out_duration(:,2)*1e3, out_duration(:,3)*1e3, out_duration(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.9290 0.6940 0.1250], 'DisplayName', 'Duration');
data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
hold on;
surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
title(ax3,'Duration')
grid on;
pbaspect([1 1 1])
xlim([-u u]) %mm
ylim([-u u]) %mm
zlim([-u u]) %mm
xlabel('X');
ylabel('Y');
zlabel('Z');
set(gca,'YDir','reverse')
str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
lgnd = legend(data1, 'Location', 'north');
lgnd.FontSize = 8;
hold off;

Min = min(Chave_time(:,6));
Max = max(Chave_time(:,6));
ax4 = nexttile;
data1 = scatter3(ax4,out_counts(:,1)*1e3, out_counts(:,2)*1e3, out_counts(:,3)*1e3, out_counts(:,4), 'filled',
'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.4660 0.6740 0.1880], 'DisplayName', 'Counts');
data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
hold on;
surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
title(ax4,'Counts')
grid on;
pbaspect([1 1 1])
xlim([-u u]) %mm
ylim([-u u]) %mm
zlim([-u u]) %mm
xlabel('X');
ylabel('Y');
zlabel('Z');
set(gca,'YDir','reverse')
str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
lgnd = legend(data1, 'Location', 'north');
lgnd.FontSize = 8;
hold off;

hlink = linkprop([ax1,ax2,ax3,ax4],{'CameraPosition','CameraUpVector'});
rotate3d on
addprop(hlink,'PlotBoxAspectRatio')

% Requires R2020a or later
fn_3dloc = append(path,'\',filenm,'_3dloc_default','.pdf');
exportgraphics(gcf,fn_3dloc,'ContentType','vector')
%% X-view
if Xview == 1
   figure('name','3d location data')
    set(gcf, 'Position',  [100, 100, 1000, 1000])
    t = tiledlayout(2,2);
    t.Padding = 'compact';
    t.TileSpacing = 'compact';

    Min = min(Chave_time(:,7));
    Max = max(Chave_time(:,7));
    ax1 = nexttile;
```

```
    data1 = scatter3(ax1,out_energy(:,1)*1e3, out_energy(:,2)*1e3, out_energy(:,3)*1e3, out_energy(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.8500 0.3250 0.0980], 'DisplayName', 'MARSE Energy');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax1,'MARSE Energy')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'ydir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    view(ax1,[-90 0]); %x
    hold off;

    Min = min(Chave_time(:,9));
    Max = max(Chave_time(:,9));
    ax2 = nexttile;
    data1 = scatter3(ax2,out_amplitude(:,1)*1e3, out_amplitude(:,2)*1e3, out_amplitude(:,3)*1e3,
out_amplitude(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0 0.4470 0.7410], 'DisplayName',
'Amplitude');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax2,'Amplitude')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,8));
    Max = max(Chave_time(:,8));
    ax3 = nexttile;
    data1 = scatter3(ax3,out_duration(:,1)*1e3, out_duration(:,2)*1e3, out_duration(:,3)*1e3,
out_duration(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.9290 0.6940 0.1250], 'DisplayName',
'Duration');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax3,'Duration')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,6));
    Max = max(Chave_time(:,6));
    ax4 = nexttile;
    data1 = scatter3(ax4,out_counts(:,1)*1e3, out_counts(:,2)*1e3, out_counts(:,3)*1e3, out_counts(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.4660 0.6740 0.1880], 'DisplayName', 'Counts');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax4,'Counts')
```

```
1437        grid on;
1438        pbaspect([1 1 1])
1439        xlim([-u u]) %mm
1440        ylim([-u u]) %mm
1441        zlim([-u u]) %mm
1442        xlabel('X');
1443        ylabel('Y');
1444        zlabel('Z');
1445        set(gca,'YDir','reverse')
1446        str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
1447        text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
1448        lgnd = legend(data1, 'Location', 'north');
1449        lgnd.FontSize = 8;
1450        hold off;
1451
1452        hlink = linkprop([ax1,ax2,ax3,ax4],{'CameraPosition','CameraUpVector'});
1453        rotate3d on
1454        addprop(hlink,'PlotBoxAspectRatio')
1455
1456        % Requires R2020a or later
1457        fn_3dloc = append(path,'\',filenm,'_3dloc_Xview','.pdf');
1458        exportgraphics(gcf,fn_3dloc,'ContentType','vector')
1459    else
1460    end
1461    %% Y-view
1462    if Yview == 1
1463       figure('name','3d location data')
1464        set(gcf, 'Position',  [100, 100, 1000, 1000])
1465        t = tiledlayout(2,2);
1466        t.Padding = 'compact';
1467        t.TileSpacing = 'compact';
1468
1469        Min = min(Chave_time(:,7));
1470        Max = max(Chave_time(:,7));
1471        ax1 = nexttile;
1472        data1 = scatter3(ax1,out_energy(:,1)*1e3, out_energy(:,2)*1e3, out_energy(:,3)*1e3, out_energy(:,4),
1473    'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.8500 0.3250 0.0980], 'DisplayName', 'MARSE Energy');
1474        data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
1475        hold on;
1476        surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
1477        title(ax1,'MARSE Energy')
1478        grid on;
1479        pbaspect([1 1 1])
1480        xlim([-u u]) %mm
1481        ylim([-u u]) %mm
1482        zlim([-u u]) %mm
1483        xlabel('X');
1484        ylabel('Y');
1485        zlabel('Z');
1486        set(gca,'ydir','reverse')
1487        str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
1488        text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
1489        lgnd = legend(data1, 'Location', 'north');
1490        lgnd.FontSize = 8;
1491        view(ax1,[0 0]); %y
1492        hold off;
1493
1494        Min = min(Chave_time(:,9));
1495        Max = max(Chave_time(:,9));
1496        ax2 = nexttile;
1497        data1 = scatter3(ax2,out_amplitude(:,1)*1e3, out_amplitude(:,2)*1e3, out_amplitude(:,3)*1e3,
1498    out_amplitude(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0 0.4470 0.7410], 'DisplayName',
1499    'Amplitude');
1500        data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
1501        hold on;
1502        surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
1503        title(ax2,'Amplitude')
1504        grid on;
1505        pbaspect([1 1 1])
1506        xlim([-u u]) %mm
1507        ylim([-u u]) %mm
1508        zlim([-u u]) %mm
1509        xlabel('X');
1510        ylabel('Y');
1511        zlabel('Z');
1512        set(gca,'YDir','reverse')
1513        str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
1514        text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
1515        lgnd = legend(data1, 'Location', 'north');
1516        lgnd.FontSize = 8;
1517        hold off;
```

```
    Min = min(Chave_time(:,8));
    Max = max(Chave_time(:,8));
    ax3 = nexttile;
    data1 = scatter3(ax3,out_duration(:,1)*1e3, out_duration(:,2)*1e3, out_duration(:,3)*1e3,
out_duration(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.9290 0.6940 0.1250], 'DisplayName',
'Duration');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax3,'Duration')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,6));
    Max = max(Chave_time(:,6));
    ax4 = nexttile;
    data1 = scatter3(ax4,out_counts(:,1)*1e3, out_counts(:,2)*1e3, out_counts(:,3)*1e3, out_counts(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.4660 0.6740 0.1880], 'DisplayName', 'Counts');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax4,'Counts')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    hlink = linkprop([ax1,ax2,ax3,ax4],{'CameraPosition','CameraUpVector'});
    rotate3d on
    addprop(hlink,'PlotBoxAspectRatio')

    % Requires R2020a or later
    fn_3dloc = append(path,'\',filenm,'_3dloc_Yview','.pdf');
    exportgraphics(gcf,fn_3dloc,'ContentType','vector')
else
end
%% Z-view
if Zview == 1
   figure('name','3d location data')
    set(gcf, 'Position',  [100, 100, 1000, 1000])
    t = tiledlayout(2,2);
    t.Padding = 'compact';
    t.TileSpacing = 'compact';

    Min = min(Chave_time(:,7));
    Max = max(Chave_time(:,7));
    ax1 = nexttile;
    data1 = scatter3(ax1,out_energy(:,1)*1e3, out_energy(:,2)*1e3, out_energy(:,3)*1e3, out_energy(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.8500 0.3250 0.0980], 'DisplayName', 'MARSE Energy');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax1,'MARSE Energy')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
```

```
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'ydir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    view(ax1,[0 90]); %z
    hold off;

    Min = min(Chave_time(:,9));
    Max = max(Chave_time(:,9));
    ax2 = nexttile;
    data1 = scatter3(ax2,out_amplitude(:,1)*1e3, out_amplitude(:,2)*1e3, out_amplitude(:,3)*1e3,
out_amplitude(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0 0.4470 0.7410], 'DisplayName',
'Amplitude');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax2,'Amplitude')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,8));
    Max = max(Chave_time(:,8));
    ax3 = nexttile;
    data1 = scatter3(ax3,out_duration(:,1)*1e3, out_duration(:,2)*1e3, out_duration(:,3)*1e3,
out_duration(:,4), 'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.9290 0.6940 0.1250], 'DisplayName',
'Duration');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax3,'Duration')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
    lgnd = legend(data1, 'Location', 'north');
    lgnd.FontSize = 8;
    hold off;

    Min = min(Chave_time(:,6));
    Max = max(Chave_time(:,6));
    ax4 = nexttile;
    data1 = scatter3(ax4,out_counts(:,1)*1e3, out_counts(:,2)*1e3, out_counts(:,3)*1e3, out_counts(:,4),
'filled', 'MarkerFaceAlpha', 0.6, 'MarkerFaceColor', [0.4660 0.6740 0.1880], 'DisplayName', 'Counts');
    data1(1).DisplayName = strcat(data1(1).DisplayName, ', n= ', string(numel(data1(1).XData)));
    hold on;
    surf(Xsample,Ysample,Zsample,'facealpha', 0.07, 'FaceColor', 'k', 'EdgeColor', 'none');
    title(ax4,'Counts')
    grid on;
    pbaspect([1 1 1])
    xlim([-u u]) %mm
    ylim([-u u]) %mm
    zlim([-u u]) %mm
    xlabel('X');
    ylabel('Y');
    zlabel('Z');
    set(gca,'YDir','reverse')
    str = {sprintf('Min. size: %.1f', Min),sprintf('Max. size: %.1f', Max)};
    text([-u -u],[(-u + 2) (-u + 2)],[(-u + 2) (-u + 1)],str,'FontSize', 9);
```

```
        lgnd = legend(data1, 'Location', 'north');
        lgnd.FontSize = 8;
        hold off;

        hlink = linkprop([ax1,ax2,ax3,ax4],{'CameraPosition','CameraUpVector'});
        rotate3d on
        addprop(hlink,'PlotBoxAspectRatio')

        % Requires R2020a or later
        fn_3dloc = append(path,'\',filenm,'_3dloc_Zview','.pdf');
        exportgraphics(gcf,fn_3dloc,'ContentType','vector')
    else
    end
    %% Other figures
    figure('name','AE characteristics (filtered)')
    set(gcf, 'Position',  [100, 100, 800, 800])
    t = tiledlayout(2,2);
    t.Padding = 'compact';
    t.TileSpacing = 'compact';

    ax1 = nexttile;
    scatter(ax1, Chave_time(:,1) ./60, Chave_time(:,7));
    title(ax1, 'Average MARSE Energy (all Ch)')
    set(gca,'yscale','log')
    xlabel('Time (min)');
    ylabel('MARSE Energy (EC#)');
    xlim([0 50])
    %ylim([30 100])

    ax2 = nexttile;
    scatter(ax2, Chave_time(:,1) ./60, Chave_time(:,9));
    title(ax2, 'Average Amplitude (all Ch)')
    xlabel('Time (min)');
    ylabel('Amplitude (dB)');
    xlim([0 50])
    ylim([40 100])

    ax3 = nexttile;
    scatter(ax3, Chave_time(:,9), Chave_time(:,6));
    title(ax3, 'Average Counts (all Ch)')
    xlabel('Amplitude (dB)');
    ylabel('Counts');
    %xlim([30 100])
    %ylim([30 100])

    ax4 = nexttile;
    scatter(ax4, Chave_time(:,8), Chave_time(:,7));
    title(ax4, 'Average Duration (all Ch)')
    xlabel('Duration (us)');
    ylabel('MARSE Energy (EC#)');
    %xlim([30 100])
    %ylim([30 100])

    % Requires R2020a or later
    fn_char = append(path,'\',filenm,'_AEchar','.pdf');
    exportgraphics(gcf,fn_char,'ContentType','vector')
    %% Histogram plots
    %Unfiltered
    if FT == 0
        figure('name','Amplitude (unfiltered)')
        set(gcf, 'Position',  [100, 100, 800, 800])
        Amp_histograms(Ch1(:,8),Ch2(:,8),Ch3(:,8),Ch4(:,8),Ch5(:,8),Ch6(:,8))
        % Requires R2020a or later
        fn_hist = append(path,'\',filenm,'_amphist','.pdf');
        exportgraphics(gcf,fn_hist,'ContentType','vector')
    elseif FT == 1
    %Filtered
        indx_Ch = find(any([Ch1(:,8) Ch2(:,8) Ch3(:,8) Ch4(:,8) Ch5(:,8) Ch6(:,8)] > 98,2));
        Ch1(indx_Ch,:) = [];
        Ch2(indx_Ch,:) = [];
        Ch3(indx_Ch,:) = [];
        Ch4(indx_Ch,:) = [];
        Ch5(indx_Ch,:) = [];
        Ch6(indx_Ch,:) = [];
        figure('name','Amplitude (filtered)')
        set(gcf, 'Position',  [920, 100, 800, 800])
        Amp_histograms(Ch1(:,8),Ch2(:,8),Ch3(:,8),Ch4(:,8),Ch5(:,8),Ch6(:,8))
        % Requires R2020a or later
        fn_hist = append(path,'\',filenm,'_amphist','.pdf');
        exportgraphics(gcf,fn_hist,'ContentType','vector')
    end
```

```
1761
1762  % AE Frequency histograms
1763  indx_Ch = find(any([Ch1(:,12) Ch2(:,12) Ch3(:,12) Ch4(:,12) Ch5(:,12) Ch6(:,12)] < 200));
1764  Ch1(indx_Ch,:) = [];
1765  Ch2(indx_Ch,:) = [];
1766  Ch3(indx_Ch,:) = [];
1767  Ch4(indx_Ch,:) = [];
1768  Ch5(indx_Ch,:) = [];
1769  Ch6(indx_Ch,:) = [];
1770  figure('name','AE Frequency')
1771  set(gcf, 'Position',  [920, 100, 800, 800])
1772  Freq_histograms(Ch1(:,12),Ch2(:,12),Ch3(:,12),Ch4(:,12),Ch5(:,12),Ch6(:,12))
1773  % Requires R2020a or later
1774  fn_hist = append(path,'\',filenm,'_freqhist','.pdf');
1775  exportgraphics(gcf,fn_hist,'ContentType','vector')
```