Seismic Project: From Signal Capture to Population Warning

Yannick Berthoumieu, Thomas Lorblanchès, Romain Tajan
yannick.berthoumieu@enseirb-matmeca.fr, thomas.lorblanches@wanadoo.fr, romain.tajan@enseirb-matmeca.fr

April 18, 2025

Abstract

The primary objective of this project is to create a MATLAB application designed to systematically dispatch rescue teams to areas affected by earthquakes and to promptly issue warnings to the population regarding potential tsunamis. Achieving this goal will require students to leverage their signal processing expertise, coding abilities, logical reasoning, and intuition.

Contents

1 Organization and Project Evaluation

This project is divided into 4 sessions of 4 hours each and 1 session of presentation. The assessment relies on a personal grade depending on the **student's behavior** (assiduity and reliability during the sessions), code quality, and a **final presentation**.

The guidelines for the final report are as follows:

- An electronic **PDF version** of the final presentation must be sent via Thor **by the day of your presentation**. The MATLAB source code produced during the project must also be sent using the same interface **at the end of yout last project session**.
- Your presentation should last **10 minutes** and should be structured as follows:
  1. **Introduction**: In this slide, you have to present the project using your **own words**. The reader should understand the goal and the context of the project.
  2. **Seismic Data Visualization**: In this section, you will address Section 3. A reader with a signal processing background should understand some key properties of seismic signals and how they relate to geological phenomena.
  3. **Automatic P-Waves Detection**: In this slide, you will address Section 4.1. The reader should understand how seismic data can be used to detect P-waves automatically. In particular, you will explain the how, what, and why of the different signal processing steps.
  4. **Numerical Results on Automatic P-Waves Detection**: In this slide, you will present some results about the method presented before. This slide is also related to Section 4.1.
  5. **Hypocenter Localization**: In this slide, you will address Section 4.2 about hypocenter localization. Your auditory should understand how the P-Wave detection step can be further processed to locate the hypocenter of an earthquake. You will also present numerical results as well as the main limitations of the proposed method and eventually how you manage to bypass those limitations.
  6. **Measuring the Intensity of the Earthquake**: In this slide, you will address Section 5. The reader should understand the process of automatically computing the magnitude of the earthquake as well as its intensity. You will also show how every feature computed during the project can be displayed to unequivocally warn the population in case of an intense earthquake or tsunami.
  7. **Conclusion**: In this slide, the reader must understand the main results of your work, and you should provide some perspective about your work.

2 Introduction

The main purpose of this project is to implement signal processing algorithms using MATLAB to detect seismic activity and analyze this activity to warn the population of a tsunami risk or dispatch rescue to areas that have experienced potentially damaging violent shaking. Timely warnings of incoming tsunamis have been demonstrated to save lives, provided they are sufficiently alerting. Therefore, ensuring both speed of issuance and accuracy of predicted height is crucial.

Before delving into data manipulation and algorithms, it is essential to provide a proper introduction to the phenomena at hand.

2.1 Earthquake and tsunami phenomena

An earthquake is a geological event which is the result of a rupture between two rock systems. This rupture starts at a given point -the hypocenter- and spreads along a surface which separates the rock systems: the fault. The orthogonal projection of the hypocenter at the surface of the Earth is called the epicenter.

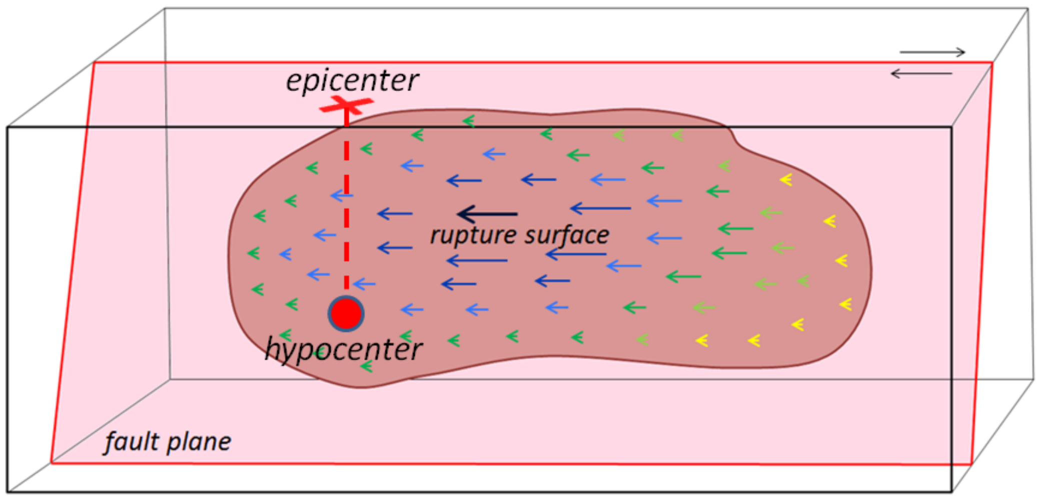

Figure 1: 3D representation of an earthquake rupture

As depicted in Figure 1, the slip amount between the two rock systems, represented by arrows, is heterogeneous along the rupture. The shaking is potentially greatest when close to a large slip zone, often assimilated to the hypocenter. However, this approximation becomes inaccurate for medium to great earthquakes with a magnitude exceeding 5.5. In such cases, as the surface area affected increases, one can be far from the hypocenter but close to the rupture zone, consequently experiencing strong shaking.

For a better understanding of the real-time unfolding of a Magnitude 7-class earthquake, you can refer to a video available at the following link: `https://youtu.be/m5xM-CTDLo0?t=60`. This footage provides valuable insight into the characteristics and progression of such seismic events.

Within the video, the area undergoing rupture (where the rocks are moving past each other) is depicted in white, while the resulting ruptured zone is highlighted in yellow. Additionally, surface seismic waves are illustrated in red.

Given the elastic properties of rock at a macroscopic scale, a rupture releases energy in the form of heat (approximately 2/3), with the remaining energy taking the form of various types of mechanical waves (around 1/3), known as seismic (or body) waves, radiating in all directions.

These seismic waves travel inside the Earth and split into two categories:

- P-Waves, which travel the fastest, reaching speeds of around $v_P = 7.3\ km.s^{-1}$ near the surface. As depicted in Figure 2, these are compression waves, similar to sound, and cause rocks to deform in the direction of propagation at a relatively high frequency (approximately 10Hz). They are also responsible for the rumbling noise experienced during an earthquake, and they typically induce moderate vertical ground shaking.

- S-Waves, which are slower (approximately 4km/s near the surface) but carry more energy than P-Waves. They can be destructive upon hitting the ground, as they induce mainly horizontal motions that buildings are vulnerable to. These waves exhibit lower frequencies (around the hertz or lower), which can resonate with the natural frequencies of structures, exacerbating their damaging effects. Additionally, S-Waves generate surface waves such as Love and Rayleigh waves upon hitting the ground above the rupture area. While these surface waves are responsible for rolling ground motion, they are beyond the scope of this discussion.

P and S waves have velocities that depend on the density of the rock, which varies with depth. This variation creates ray patterns that curve inside the Earth due to gradients in the refraction coefficient.

For the remainder of this subject, since we are dealing with earthquakes at close range, we make the assumption of a constant velocity model with straight rays.

You can find a short introductory video on the wave propagation patterns at this link `https://www.iris.edu/hq/inclass/animation/3component_seismogram_records_seismicwave_motion`.

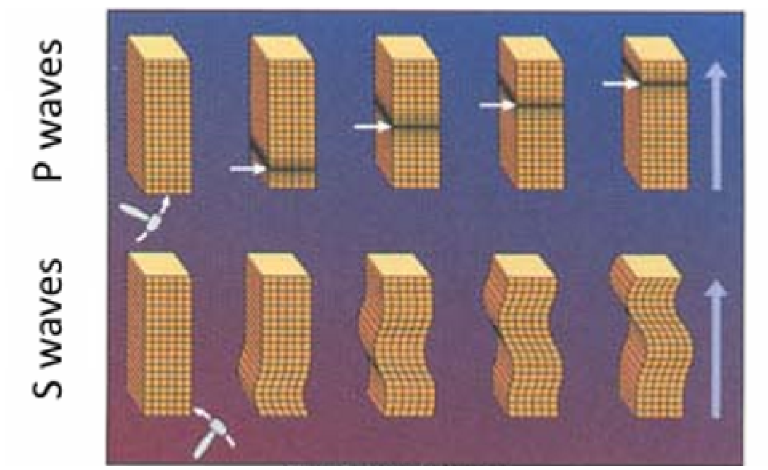

Figure 2: P and S waves propagation patterns

When an earthquake occurs underwater, with a rupture area close to the seafloor and a mean slip vector (which would be the average of all the arrows in Figure 1) having a significant vertical component, there is a risk of a tsunami. As depicted in Figure 3, the vertical displacement between the two rock blocks lifts up (or pushes down) a considerable amount of water.

Tsunamis travel at approximately 800 km/h in deep water, decelerating to around 50 km/h but increasing in height as they approach the coast. Their destructive power lies in the fact that, unlike typical wind-induced waves, the energy they carry is distributed throughout the entire water column, rather than being concentrated near the surface. Additionally, their wavelengths can exceed dozens of kilometers.

This results in a series of rapid and comprehensive inundations of coastlines, with the first wave often not being the largest.

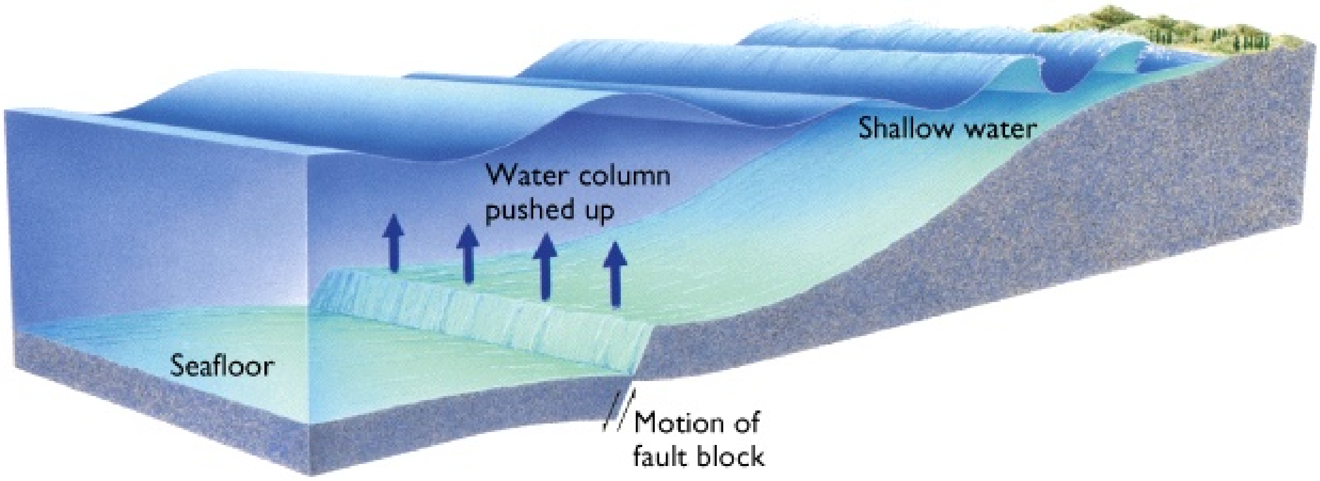

Figure 3: Tsunami generation mechanism

For these reasons, tsunami warning systems have been developed around the world. The objective is to deliver accurate warnings as quickly as possible. For instance, the Japan Meteorological Agency can issue a tsunami warning just 3 minutes after the occurrence of an earthquake.

2.2 Goal of this project

The main purpose of this project is to implement signal processing algorithms using MATLAB to detect seismic activity and analyze the collected data to provide early warnings for tsunami risks and inform about the felt effects. Throughout this project, the following objectives will be pursued:

- Automatically detect the occurrence of an earthquake using signal processing algorithms.
- Automatically locate the hypocenter of an earthquake in space and time through data analysis techniques.
- Automatically evaluate the magnitude of an earthquake using appropriate methodologies.

A tool for warning people in case of a tsunami hazard will be developed based on the outcomes of these objectives.

2.3 Signal acquisition

Seismic activity is monitored thanks to seismographs that record ground motion in different Japanese cities. Each station records a three-dimensional acceleration signal $\mathbf{a}(t) = [a_x(t)\ a_y(t)\ a_z(t)]$ at a sampling frequency of $F_s = 100$ Hz, where

- a_x is the acceleration in the North-South direction,
- a_y is the acceleration in the East-West direction,
- a_z is the acceleration in the Up-Down direction.

In the MATLAB data, $\mathbf{a}(t)$ is given as a $3 \times N$ matrix where the first column represents $a_x(t)$, the second represents $a_y(t)$, and the third represents $a_z(t)$.

2.4 Signal Database

A database containing different earthquake recordings from various stations in Japan is provided. It can be downloaded at `https://rtajan.github.io/ts114/`. Each .mat file can be loaded into the `Matlab` workspace using the command `load`, and is composed of a vector of structures called *list*. More specifically, the s^{th} element of this vector (`list(s)`) is itself a structure containing the following fields:

- `list(s).sismo` contains a three-dimensional acceleration signal in cm/s^2. The three components of this field represent acceleration in the North-South direction, the East-West direction, and the Up-Down direction.
- `list(s).id` contains the ID of the station (as well as its name in Japanese).
- `list(s).lon` is the longitude of the station.
- `list(s).lat` is the latitude of the station.
- `list(s).offset` is the time offset (in seconds) of the first sample of the signal (this offset is given compared to a common reference for all signals).

3 Signal Visualization

The goal of this section is to display seismic signals and highlight their main physical properties. The purpose for an engineer is to understand how these properties can be exploited inside automatic processes to extract features such as the earthquake distance or position. In a report, figures are supposed to highlight some key concepts, hence they must be presented, labeled, and have a caption indicating what should be observed in the figure.

3.1 Time display

Matlab. *Under Matlab, load the signals given in the file entitled recording_ 1.mat and display the time evolution of the* 5^{th} *recording. On the same figure, highlight regions of this signal containing sensor noise (when there is no earthquake-induced shaking), the P-wave, and the S-wave.*

Report task. *In the section dedicated to the seismic signaling, use the previous Matlab task to illustrate the following key points presented in introduction about seismic waves*

- *S-waves are more powerful than P-waves,*
- *P-waves predominantly propagate vertically,*
- *S-waves predominantly propagate horizontally.*

A suggestion to answer this task is to present those figures:

- A figure of the complete signal that you can annotate.
- A "zoom" focusing on the very beginning of the earthquake.
- A "zoom" focusing on the S-wave.

3.2 Time/Frequency display

As seismic signals are non-stationnary, we will then use a spectrogram to display their frequency content. Indeed, seismic activity varies slowly compared to the sampling frequency of $100Hz$ and the P-wave comes first then the S-Wave. The Fourier transform (or Discrete Fourier Transform) provides the frequency content of the signal, however it does not show the variation of the signal through time well. Consequently, we analyze $x(t)$ signal using a sliding window. In particular, computing the Discrete Fourier Transform (DFT) of each window captures the frequency content of the windowed signal and how it changes over time. This operation is called **Short Term Fourier Transform** (STFT). Mathematically, the STFT is expressed as follows

$$X(m,\nu) = \sum_{n=0}^{L-1} x_n w_{n-m} e^{-j2\pi\nu n}. \tag{1}$$

One can show that the computation of the STFT for $m = 0, d, 2d, \ldots$ can be performed following these three steps:

1. decompose the vector x (of length L) into $M = \lfloor \frac{L}{d} \rfloor$ frames of N samples. Each frame will be stored as columns. The first frame contains elements $(x_0, \ldots x_{N-1})^T$. The second frame contains elements $(x_d, \ldots x_{d+N-1})^T$ so that it overlaps the first frame on the $N-d$ last elements. Figure 4 illustrates this processing.

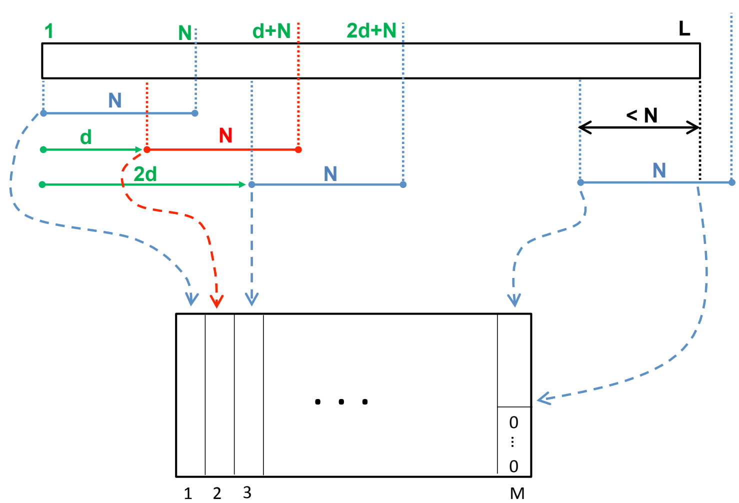

Figure 4: Framing of x_n.

2. multiply each column of the matrix created in step 1 by a window $w = (w_0, w_1, \ldots w_{N-1})$. The window signal w_n can be rectangular, triangular, Hamming, or Hanning, however in this work, we will only consider rectangular or Hamming window. Figure 5 illustrates this processing.

3. Compute the DFT on N_{fft} points of each column using `fft`. We recall that if x has a length N, `fft(x)` computes the DFT at the frequencies $\nu = (0, \frac{1}{N}, \ldots \frac{N+1}{N})$, where $\nu = \frac{f}{F_s}$. On the other hand, if x has a length N, `fft(x,Nfft)` computes the DFT at the frequencies $\nu = (0, \frac{1}{N_{fft}}, \ldots \frac{N_{fft}-1}{N_{fft}})$, where $\nu = \frac{f}{F_s}$

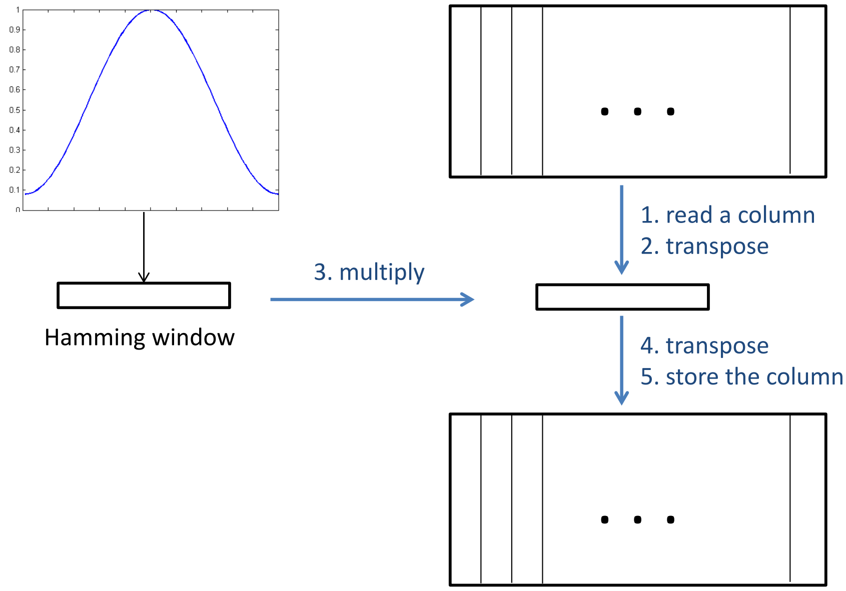

Figure 5: Windowing of x_n.

Matlab. *Create a function* `stft` *that performs the STFT of the signal x_n. This function will have the following header*

stft.m
```
function [X, f, t] = stft(x,w,d,N_fft,Fs)
% This function computes the stft for m = [0, d, 2d, 3d...]
% This function outputs are:
% -> X, which is  a matrix of n_fft lines and M columns
%    M is the number of elements of m
%    X(i,j) is the value of the STFT for time t(i) and frequency f(j)
% -> f, is a column vector of the frequencies (in Hz)
% -> t, is a row vector containing the times of the beginning of the windows
```

The modulus squared of the STFT is called the spectrogram

$$S_x(m,\nu) = \frac{1}{N}\left|X(m,\nu)\right|^2. \tag{2}$$

Matlab. *Create a function* `spectro` *that computes the spectrogram of x_n.*

spectro.m
```
function [Sx, f, t] = spectro(x,w,d,N_fft,Fs)
% This function computes the spectrogram for m = [0, d, 2d, 3d...]
% This function outputs are:
% -> Sx, which is  a matrix of n_fft lines and
%                             M (number of elements of m) columns
%    Sx(i,j) is the value of the spectrogram for time t(i) and frequency f(j)
% -> f, is a column vector of the frequencies (in Hz)
% -> t, is a row vector containing the times of the beginning of the windows
```

MATLAB. *Under MATLAB, load the signals given in the file entitled recording_ 1.mat and display the spectrogram of the* 5^{th} *recording. On the same figure, highlight regions of this signal containing background noise (when there is no earthquake), the P-wave, and the S-wave.* ***At first, you can consider a rectangular window of*** 1 ***second (***100 ***samples) with an overlap of*** 99 ***samples.*** *From this first spectrogram, you can now play with the different parameters, for example:*

- *you can change the type of the window: Hamming, Hann, Blackman, Welsh...*
- *you can change the duration of the window (*$0.5s$ *to* $5s$ *or more)*

Report task. *Choose a spectrogram in which the P-Wave and the S-wave can easily be observed and comment this spectrogram in your report.*

MATLAB. *After computing the spectrogram, this task is divided into two sub-tasks:*

1. *Compute the energy contained in each window of the spectrogram (Parseval's identity can be used). Plot this energy as a function of time.*
2. *Compare the spectrum of a window of the signal containing the P-wave only with another window containing the S-wave.*

Report task. *From the previous tasks, illustrate the fact that S-waves are much more powerful than P-waves and observe the frequency content of both type of waves.*

4 P-Wave detection and epicenter localization

4.1 P-Wave detection

For station s, the P-wave can be automatically detected using the algorithm described hereafter. For each sample n in the signal of station s, compute the following two signals :

$$STA^{(s)}(n) = \frac{1}{N_l} \sum_{i=n}^{n+N_l-1} \left|a_z^{(s)}(i)\right|^2 \tag{3}$$

$$LTA^{(s)}(n) = \frac{1}{N_L} \sum_{i=n-N_L}^{n-1} \left|a_z^{(s)}(i)\right|^2 \tag{4}$$

where N_l and N_L are integers respectively equivalent to duration $T_l = 1$ and $T_L = 5$ seconds. The start of the P-Wave is then located by finding the **smallest** n such that the ratio of STA and LTA is above a threshold Γ, that is

$$n_P^{(s)} = \inf \left\{ n : \frac{STA^{(s)}(n)}{LTA^{(s)}(n)} > \Gamma \right\} \tag{5}$$

A good value for Γ can be $\Gamma = 2.5$ but feel free to try other values. The time in seconds at which the station s can detect a P-wave is then obtained by converting the result of equation (5) into seconds :

$$t_P^{(s)} = t_O^{(s)} + \frac{n_P^{(s)} - 1}{Fe} \tag{6}$$

where $t_O^{(s)}$ is a timing offset.

MATLAB. *Implement a function* `sta_lta` *that takes as input* $a_z^{(s)}(t)$, F_e, T_l, T_L, Γ, $t_O^{(s)}$ *and returns* $t_P^{(s)}$.

Report task. *Explain the STA/LTA algorithm with your own words. In particular, you should lay emphasis on*

- *why this algorithm works, what is the rationale behind this method*
- *which signal processing steps you used to implement this algorithm (filters, FFT, ...)*
- *what happens if* Γ *is changed*

(Hint) If illustrations about this method are deliberately removed, consider adding some. You could draw the algorithm as a signal processing chain and plot the signals to enhance understanding.

4.2 Estimation of the hypocenter localization

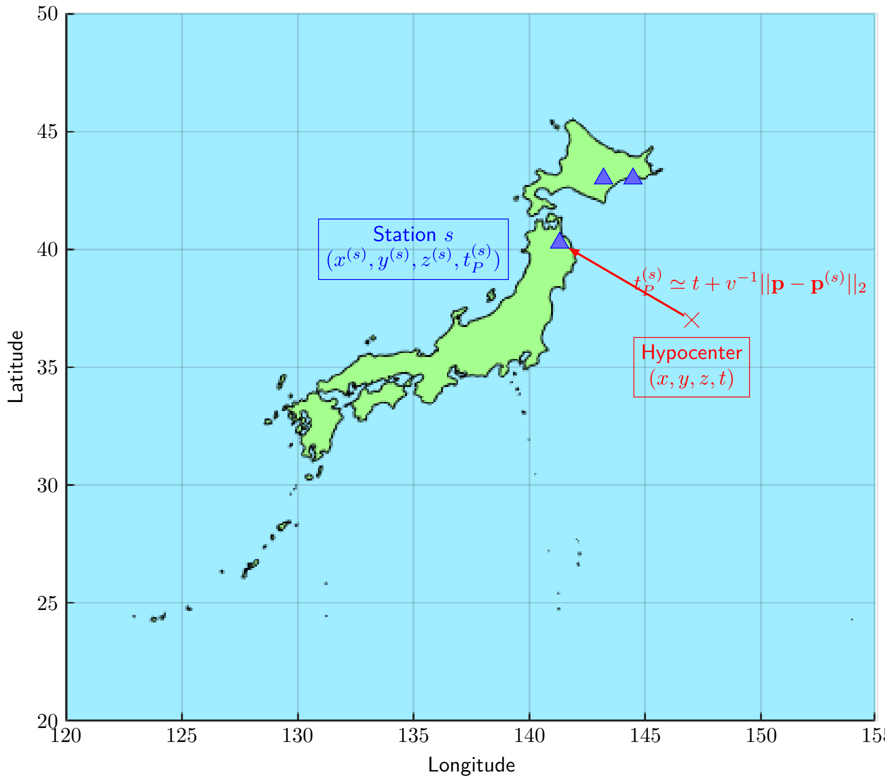

Figure 6: Example of hypocenter estimation

The algorithm for locating the hypocenter is as follows:

1. For each station s in the database, compute the measured time of arrival of the P-wave using STA/LTA. Let $t_P^{(s)}$ represent this measured time of arrival.
2. For each station s in the database, compute its Cartesian coordinates from its longitude and latitude using an equirectangular projection centered around a relevant point of your choice. This projection and its associated center will be the same for all stations. Let the vector $\mathbf{p}^{(s)} = (\mathbf{x}^{(s)}, t_P^{(s)}) = (x^{(s)}, y^{(s)}, z^{(s)}, t_P^{(s)})$ represent this position in space and time. We will assume $z^{(s)} = 0$.

3. Let $\mathbf{p} = (\mathbf{x}, t)$ represent the position (in space and time) of the hypocenter at the onset of the earthquake. We will use a simple model for the P-waves: they are assumed to travel straight through the Earth at a speed of $v_p = 7300$ m/s. The modeled time $\hat{t}_P^{(s)}$ at which the P-wave reaches station s can then be expressed as:

$$t_P^{(s)} \simeq t + v_p^{-1}||\mathbf{x} - \mathbf{x}^{(s)}||_2 \tag{7}$$

The hypocenter is then located by finding $\mathbf{p}$ that provides the best fit between the measured arrival times and those generated by our model using $\mathbf{p}$. This is expressed as finding $\mathbf{p}$ that minimizes the sum of quadratic errors between the modeled and measured times of arrival:

$$J(\mathbf{p}) = \sum_{s=1}^{S} (\underbrace{t - t_P^{(s)} + v_p^{-1}||\mathbf{x} - \mathbf{x}^{(s)}||_2}_{\Phi^{(s)}(\mathbf{p})})^2 \tag{8}$$

This function will be iteratively minimized using a damped Gauss-Newton procedure.

MATLAB. *1. Develop a* MATLAB *function* `phi` *that accepts the vector* $\mathbf{p}$ *and the reference positions* $(\mathbf{p}^{(s)})_s$ *as inputs, and returns the vector*

$$\boldsymbol{\Phi}(\mathbf{p}) = \left(\Phi^{(1)}(\mathbf{p}), \ldots, \Phi^{(S)}(\mathbf{p})\right)$$

.

2. *Design a* MATLAB *function* `Jphi` *to calculate the Jacobian matrix* $J_{\boldsymbol{\Phi}}(\mathbf{p})$ *from an input vector* $\mathbf{p}$ *and from the reference positions* $(\mathbf{p}^{(s)})_s$.

3. *Utilize functions (a) and (b) to create a* MATLAB *function* `gauss_newton` *for iterative computation of the hypocenter position, governed by the equation:*

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \alpha(J_{\boldsymbol{\Phi}}^T J_{\boldsymbol{\Phi}})^{-1} J_{\boldsymbol{\Phi}}^T \boldsymbol{\Phi}(\mathbf{p_n})^T$$

where α *denotes a damping factor, typically set to* 0.99, *and* $J_{\boldsymbol{\Phi}} = J_{\boldsymbol{\Phi}}(\mathbf{p_n})$.

Report task. *First, initialize* $\mathbf{p}_0$ *corresponding to a point with longitude* 146, *latitude* 36, *altitude* -10 *km, and initial time* 0*s.*

Regarding the figure illustrating the cost function $J(\mathbf{p}_n)$ *as a function of* n *for* $\alpha = 0.01$ *and* 0.1, *we observe the convergence behavior of the Gauss-Newton algorithm for different damping factors. A smaller damping factor (*$\alpha = 0.01$*) leads to slower but potentially more stable convergence, while a larger damping factor (*$\alpha = 0.1$*) may result in faster convergence but with a risk of oscillations or instability.*

Finally, compare the final values obtained from the algorithm with the ground truth:

- *Latitude:* 38.20
- *Longitude:* 141.92
- *Time:* 3.4 *s*
- *Depth:* 66 *km*
- *Local Magnitude:* 7.2

Report task. *For* $\alpha = 0.1$, *comment the result obtained for the following initial points :*

- $\mathbf{p}_0$ *of longitude* 130*, latitude* 25*, altitude* $-10km$ *and initial time* $0s$*.*
- $\mathbf{p}_0$ *of longitude* 146*, latitude* 36*, altitude* $10km$ *and initial time* $0s$
- $\mathbf{p}_0$ *under the first triggered station, altitude* $-10km$ *and initial time* $0s$

Propose a way to solve these problems.

To address these problems, consider employing a multi-stage optimization approach. Start with a coarse grid search to approximate the initial position, followed by a more refined optimization method such as the Gauss-Newton algorithm. Additionally, incorporating more stations or additional seismic data could improve the accuracy of the results.

5 Estimation of the earthquake impact

Several tools exist to assess the potential earthquake impact on a populated area among which one can find those described in the following subsections.

5.1 JMA Seismic Intensity

Seismic intensity scales are designed to characterize damage and human perception of shaking at a point on the ground surface (distinct from magnitude). Each earthquake generates an uneven spatial intensity distribution. Generally, the closer you are to a ruptured area, the higher the seismic intensity. Additionally, local soil composition can amplify shaking and intensity.

Among the widely used seismic intensity scales are the Modified Mercalli Intensity (MMI), EMS98 (used in Europe), etc.

The Japan Meteorological Agency (JMA) developed its own seismic intensity scale, also called the Shindo scale. It categorizes local seismic intensity into 10 levels, which, in terms of human perception, can be summarized as follows:

- 0 ($I_{JMA} < 0.5$): Imperceptible to people but recorded by seismometers.
- 1 ($0.5 \leq I_{JMA} < 1.5$): Slightly felt by some people keeping quiet in buildings.
- 2 ($1.5 \leq I_{JMA} < 2.5$): Felt by many people keeping quiet in buildings. Some people may be awakened.
- 3 ($2.5 \leq I_{JMA} < 3.5$): Felt by most people in buildings. Felt by some people walking. Many people are awakened.
- 4 ($3.5 \leq I_{JMA} < 4.5$): Most people are startled. Felt by most people walking. Most people are awakened.
- 5 Lower ($4.5 \leq I_{JMA} < 5.0$): Many people are frightened and feel the need to hold onto something stable.
- 5 Upper ($5.0 \leq I_{JMA} < 5.5$): Many people find it hard to move; walking is difficult without holding onto something stable.
- 6 Lower ($5.5 \leq I_{JMA} < 6.0$): It is difficult to remain standing.
- 6 Upper ($6.0 \leq I_{JMA} < 6.5$): It is impossible to remain standing or move without crawling. People may be thrown through the air.
- 7 ($I_{JMA} \geq 6.5$): Same as 6 Upper for human perception, causing great damage to structures.

For more details about this seismic scale, visit: `https://www.jma.go.jp/jma/en/Activities/inttable.html`.

This seismic intensity scale is computed directly from the norm of a filtered acceleration vector defined as:

$$a(t) = ||\mathbf{a}_f(t)||_2$$

First, each component of $\mathbf{a}_f(t)$ is filtered using a bandpass filter with coefficients provided at `https://rtajan.github.io/assets/cours/TS114/jma_filter.mat`. Then, the norm is calculated. The output of this process, $a(t)$, is further processed to compute the JMA instrumental seismic intensity scale I_{JMA}. These steps are detailed in the section *JMA Instrumental Seismic Intensity* of [1].

Matlab. *After reviewing the relevant section in [1], you can write code to compute the JMA seismic intensity level for a given station.*

Report task. *Detail the algorithm for the signal processing steps you have developed to compute the JMA seismic intensity. Subsequently, present the results of your algorithm on the map of Japan. Each station should be represented using a color indicating the JMA intensity level. Your final map should resemble the one available at:* `https://www.jma.go.jp/en/quake/20200518030442395-18120046.html`*.*

5.2 Magnitude

The magnitude of an earthquake serves as an indicator for comparing earthquakes and provides insights into the extent of the ruptured area. Various definitions of magnitude exist, two of which are discussed below.

5.2.1 The Local Magnitude M_L

Introduced by Charles Francis Richter in 1935 as the Richter Scale and subsequently updated, it offers the advantage of providing a quick estimation of earthquake size, often used for Early Warning systems. It operates under the assumption that the earthquake is a point source with instantaneous rupture, correlating the energy released with maximum displacement amplitude and an attenuation formula. However, this assumption causes the scale to saturate for magnitudes exceeding 8.0, limiting its ability to distinguish between large and very large events.

The local magnitude M_L is computed as the empirical mean of magnitude estimations at each seismic station $m_l^{(s)}$ within a 200 km radius from the hypocenter. Let $\mathcal{S}$ denote the subset of stations for which $D^{(s)} = ||\mathbf{x} - \mathbf{x}^{(s)}||_2 \leq 200$ km:

$$M_L = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} m_l^{(s)} \tag{9}$$

The magnitude estimation $m_l^{(s)}$ at a given station s is described as:

$$m_l^{(s)} = 1.1235 \times \left[\log_{10}(\Delta_{\max}^{(s)}) + \log_{10}(D^{(s)}) + 1.9 \times 10^{-3} D^{(s)} - 5 \times 10^{-3} d_H - 0.02\right] \tag{10}$$

Where:

- $\Delta_{\max}^{(s)}$ is the maximum modulus of high-pass filtered displacement of the station at a 0.5 Hz cut-off frequency, in μm. The displacement along an axis is obtained by integrating the acceleration along this axis twice. You can use the Matlab function `cumtrapz` to integrate and use `[a,b]=butter(2,0.5/Fe,'high')` to design the high-pass filter (a and b as numerator and denominator for the `filter` function).

- $D^{(s)}$ is the distance between the station and the hypocenter in km.
- d_H is the hypocenter depth in km.

Matlab. *Write Matlab code to compute the local magnitude M_L of the earthquake. Compare it with the value given in Section 4.2.*

5.2.2 The Moment Magnitude M_W

Introduced by Thomas Hanks and Hiroo Kanamori in 1979, the Moment Magnitude is globally adopted as the definition for significant earthquakes ($M_W \geq 6.0$). It is incorrect to refer to it as "Richter Magnitude". The Moment Magnitude evolves logarithmically with the total amount of seismic energy released by the fault rupture, giving it a physical meaning. Its computation requires heavy calculations, taking several minutes on supercomputers, making it unsuitable for Early Warning systems. This is because it mathematically reconstructs the rupture surface and displacement distribution from the inversion of the actual seismic waves.

However, since it is the best estimator even for sizable earthquakes, its value is required for issuing reliable Tsunami Warnings.

5.3 Tsunami Alert

As discussed in the introductory part, Tsunami Warnings are crucial for alerting people about incoming tsunamis. These warnings are typically divided into several levels based on the expected height h of the tsunami waves, which determines the expected population response:

- Tsunami Advisory: 50 cm $\leq h < 1$ m. The population is advised to leave beaches and ports.
- Tsunami Warning: 1 m $\leq h < 3$ m. The population is instructed to evacuate the coast and riversides to designated areas.
- Major Tsunami Warning: $h \geq 3$ m. The population must evacuate the coast and riversides to higher designated areas as fast as possible or to concrete buildings higher than 3 stories.

Modern methods utilize rupture process modeling to compute the distribution and orientation of fault rupture, thereby estimating the displaced amount of water. However, this requires heavy computation even for supercomputers. To simplify the process, a common practice is to compute the local magnitude M_L and convert it into an approximated moment magnitude $M_{W'}$ using the following empirical formula:

$$M_{W'} = M_L - 0.171 \tag{11}$$

For simplicity, in this work, we use this magnitude to predict the wave heights using spheres centered on the hypocenter. The radius in kilometers of the spheres corresponding to each warning level height is given by:

$$\begin{aligned} r_{0.5} &= 4.45 \times 10^{M_{W'} - 5.85} \\ r_1 &= 3.78 \times 10^{M_{W'} - 6.21} \\ r_3 &= 4.05 \times 10^{M_{W'} - 6.59} \end{aligned}$$

To display a warning radius on a map, the radius of the circle described by the intersection between the warning sphere and the Earth's sphere is used.

Note that the intersection exists only if the sphere radius is greater than the depth d of its center. In this case, the intersection radius $\bar{r_h}$ is given by:

$$\bar{r_h} = R \arccos\left(\frac{R^2 + (R-d)^2 - r_h^2}{2R(R-d)}\right) \tag{12}$$

Where:

- R is the Earth radius (6371 km)
- d is the hypocenter depth in km (positive value)
- r_h is the warning sphere radius in km for the wave height h

Matlab. *Implement a function* `tsunami_warning` *which takes the hypocenter 3-D Cartesian coordinates and local magnitude* M_L *as input and returns the intersection radius of each warning level (*$\bar{r_{0.5}}$*,* $\bar{r_1}$*, and* $\bar{r_3}$*). Output zero values as default if the warning sphere radius is smaller than the hypocenter depth.*

Report task. *Plot the epicenter and warning circles for each warning level. Determine the actions the population should take based on the warning levels.*

6 Bonus

Matlab. *Transform your main script into a function* `population_warning` *which takes a station dataset as input and outputs the earthquake latitude, longitude, depth, local magnitude, relative time of occurrence, as well as figures of intensity map and tsunami warning circles.*

Report task. *Run the function on the other provided datasets. What observations can you make about the fourth earthquake? Its real moment magnitude* M_W *was as high as* 9.1*. Discuss why the algorithm could have underestimated it and the potential impact on populations.*

References

[1] Khosrow T Shabestari and Fumio Yamazaki. A proposal of instrumental seismic intensity scale compatible with mmi evaluated from three-component acceleration records. *Earthquake Spectra*, 17(4):711–723, 2001.

[2] Thomas C. Hanks and Hiroo Kanamori. A moment magnitude scale. *Journal of Geophysical Research: Solid Earth*, 84(B5):2348–2350, 1979.