Institute of Geosciences | Geophysics | Seismology | Volcanology

MuRAT

Multi-Resolution seismic Attenuation Tomography

Version 3.0

Luca De Siena

Associate Professor of Geophysics with Volcanology at JGU Mainz, Germany

Honorary Lecturer at the University of Aberdeen, UK

work phone: +49 (0) 61313928499

cell: +49 (0) 15122316065

email: ldesiena@uni-mainz.de

internet: www.lucadesiena.com

MuRAT in Brief

MuRAT is a Matlab Package for Seismic Attenuation, Scattering and Absorption Tomography using Body and Coda Waves. The quickest way to implement MuRAT is to read his online readme and follow step-by-step his wiki. This PDF document remains the complete manual for MuRAT.

Features

What it does: *MuRAT* is a software package for measuring seismic attenuation, scattering, and absorption from passive and active data and model 3D variations of these parameters in space.

The first MuRAT: *MuRAT1.0* was developed by Luca De Siena during his PhD at the INGV-Osservatorio Vesuviano, Italy, and ***published in 2014*** while he was research assistant at the Westfälisches Wilhelms Universität, Münster. Three sample papers published using this code are: ***De Siena et et al. 2017, SR***, ***Prudencio et al. 2020***, and ***Sketsiou et al. 2021***;

In 2D: ***MuRAT2D*** is the result of the activity of the ***Volcano Earth Imaging group***. It produces 2D seismic scattering and absorption maps at multiple frequencies and is suited for largely undetermined problems. Three example papers published using this code are: ***De Siena et et al. 2017, GRL***, ***Napolitano et al. 2020***, and ***Sketsiou et al. 2020***;

In 3D: ***MuRAT3.0 (or MuRAT3)*** is the current standard for measuring total attenuation, scattering and absorption - the project started in 2021. Reference papers are ***Reiss et al. 2022*** and ***Gabrielli et al. 2023***. This guide describes this code.

Contents

Chapter 1

Introduction

This document describes how to use the Matlab software package MuRAT to perform seismic tomography: however, it does not follow the traditional path of imaging *seismic velocity* with travel times or *phase/group velocities* using surface waves. MuRAT is a code for *seismic attenuation tomography*.

To understand the difference, consider the following seismogram produced by a volcanic earthquake:

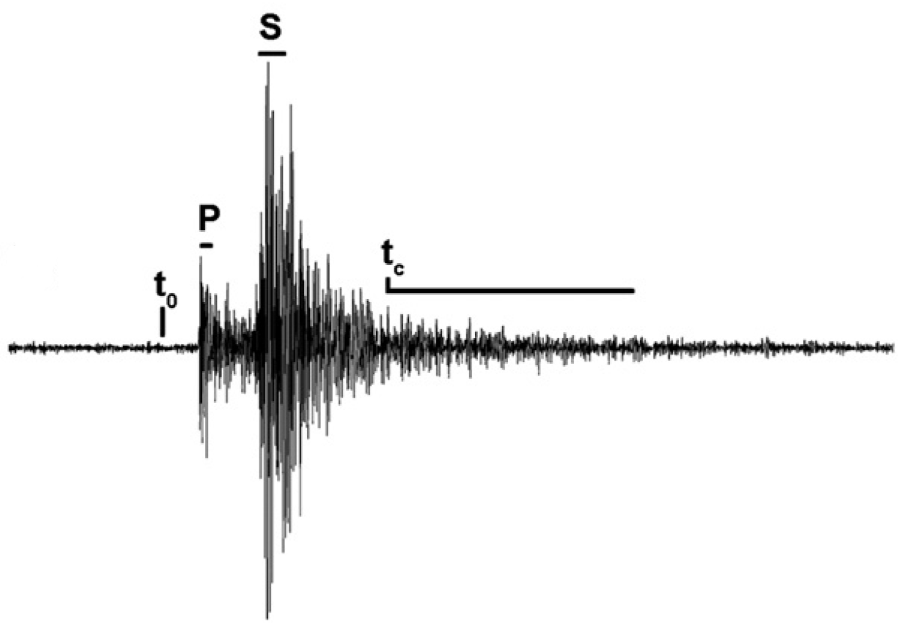


Figure 1.1: A seismogram. MuRAT measures the heterogeneous loss of seismic energy using different seismic attributes

We are used to picking phases on it - for example, the P-wave arrival. We measure velocity changes over different paths by modelling the source-station path followed by the corresponding wave packet with a *ray-tracing* approach. Travel-time tomography requires a *non-linear inversion*, where we must jointly update the paths followed by the waves, the corresponding velocity model, and seismic locations to find a reliable solution. Multiple codes do this: our favourites are LOTOS and FMTOMO (all underlined names in the manual are links to the related internet sites). However, MuRAT is part of a more comprehensive framework working across multiple languages (Matlab, Python and Julia). This comprises codes that allow the user to measure and image body-wave and surface wave velocity in space (from earthquake and noise) (interfaced with python) and full waveform inversions (in Julia). This framework especially couples seismology with outputs from Geodynamics codes, like LaMeM, efficiently. Combining MuRAT with the partner codes is highly recommended for complete imaging and modelling of the area under study.

MuRAT deals with the energy loss seismic waves suffer while propagating through the heterogeneous Earth. It fills the gap of an open-access tool to measure attenuation, separate the processes that lead to it, and model them in space. To our knowledge, the first mention of an *attenuation tomography* was given by Ho-Liu et al. (1988) in the Coso Valley. More than 30 years afterwards, this technique is standard with global, regional and local applications. Yet, different mechanisms attenuate seismic waves while they propagate. While codes to model total attenuation in space exist, no open-access code can measure attenuation, separate the mechanisms that produce it and model their variations in space with seismic inversion. No code can undoubtedly do it using as input seismograms downloaded from the data servers.

We define here the mechanisms causing seismic attenuation:

1. Seismic energy progressively spreads over larger volumes while propagating far from the source. The energy thus decreases with distance due to *geometric spreading*. This mechanism is independent of frequency for a given phase. However, depending on frequency, often more than one phase is included in a *window of time* measured on a seismogram - in Fig. 1.1, it is challenging to distinguish between S-waves and surface waves produced right after them.

2. Seismic energy is lost into the heterogeneous Earth depending on the ratio between the wavelength (λ) and the dimension of the heterogeneity (a). The energy thus decreases with distance due to *scattering losses*. This mechanism depends on the frequency and comprises complex physics, as those described by multi-pathing and diffractions. Reflection and refraction can be characterised as scattering processes.

3. Seismic energy interacts with the heterogeneous Earth losing heat for each cycle. The energy thus decreases with distance due to *absorption losses* in the materials they pass through. This mechanism depends on frequency. How much energy the Earth absorbs is challenging to quantify, but such a quantification allows connecting seismology with other disciplines.

4. The sum of scattering and absorption losses gives *total attenuation*. This is the attenuation experienced by direct wave packets while they propagate through the heterogeneous Earth. When we talk about seismic attenuation, we generally refer to total attenuation. However, this is only valid in specific circumstances and for particular phases.

MuRAT measures and models total attenuation, scattering attenuation and absorption in space using single seismogram observations. The first three chapters of this documentation offer an overview of the theory underlying attenuation tomography in these three forms.

In chapter 2, we revise the theory behind measurement and modelling of the energy lost by direct waves (**P** and **S**, Fig. 1.1). The method used to model total attenuation in chapter 2 is the *coda normalisation* (CN) method (Aki, 1980; Yoshimoto et al., 1993). In tomography, the method has been introduced by Del Pezzo et al. (2006). At the core of MuRAT is this seminal paper, which has been developed by De Siena et al. (2009) and De Siena et al. (2010) for muti-scale applications. De Siena et al. (2010) benchmark the CN with the standard spectral slope (t^*) method, applied by the global community. For a decade, the method has only been applied to volcanoes. The reason is the need for a relevant *coda* (t_c, Fig. 1.1) to normalise direct-wave information. The method has the advantage of removing source and site effects from direct wave measurements but suffers from an inexact description of the process. MuRAT uses the standard description of geometric spreading, r^α, with r being the hypocentral distance and α the geometric spreading coefficient, measuring α with its uncertainty. By comparing the results to the assumed value of α for direct wave energies ($\alpha = 1$), the user knows how far this is from the theory. MuRAT3D deals with the problem associated with direct-wave measurements and modelling, using theoretical and computational results obtained throughout the last decade.

In chapter 3, we revise the theory behind the use of *peak delays* as markers of scattering losses (Saito et al., 2002; Takahashi et al., 2007, 2009; Tripathi et al., 2010; Calvet and Margerin, 2013; Calvet et al., 2013; De Siena et al., 2016; Sato, 2016). Peak delays are a focus of the early-warning community and a quantity important to seismic source modelling in the near field. The standard theory recognises peak delay as a marker of scattering losses in the far field, modelled by the Markov approximation (Saito et al., 2002), in the case scattering is produced only by long wavelength component ($\lambda << a$. However, recent research also suggests a dependence of peak delay on scattering when λ is of the order of propagation distance and correlation length of heterogeneity (Napolitano et al., 2020; Di Martino et al., 2021). Here, phenomena like diffraction and trapping take over. The chapter discusses the standard application, theoretical limitations and future outlooks.

In chapter 4, we revise the theory behind the use of *late-time coda attenuation* as marker of seismic absorption (Wegler and Luehr, 2001; Prudencio et al., 2013; Calvet and Margerin, 2013; Del Pezzo et al., 2016; De Siena et al., 2016, 2017a; Del Pezzo et al., 2018; Sketsiou et al., 2020). Its application relies on a preliminary Multiple-Lapse-Time-Window-Analysis (MLTWA - Fehler et al. (1988)). Coda attenuation can be described using single and multiple scattering. MuRAT provides the tools to test the onset of equipartition, a necessary condition for diffusion, where coda attenuation is equivalent to seismic absorption. The embedded sensitivity kernels model coda attenuation in space based on Paasschens equations (Paasschens, 1997) and the approximations of Pacheco and Snieder (2005). The theory is inexact in the case of sharp vertical or lateral boundaries, so much so that diffusion could never onset Morozov (2008). However, in zones where the Moho is thick and for sufficient data, the theory generally produces stable results (Sketsiou et al., 2020). New work has been recently developed to include the effect of sharp boundaries and vertical velocity variations on coda attenuation measurements (Sanborn et al., 2017; Cormier and Sanborn, 2019; Nardoni et al., 2021). MuRAT currently provides the tool to invert for coda attenuation using a diffusive approximation and sensitivity kernels.

From chapter 5, this manual describes how to install and use the code, so you can start from there if you already know the theory.

Chapter 5 describes installation procedures and online documentation tools for MuRAT. MuRAT 3.0 and its Readme.txt are available on GitHub and have a Wiki explaining the primary steps to run the code and interpret results. We discuss why the code is released in proprietary (Matlab©) language and the corresponding installation procedures.

Chapter 6 describes data preparation for MuRAT. MuRAT 3.0 accepts only Seismic Analysis Code (SAC) data. SAC is a standard for seismologists and the favourite format when converting miniseeds on data servers like IRIS. You may have been processing data with another software. This chapter will discuss the conversions necessary to create a readable dataset for MuRAT, especially the tools you can use to assess if the data are in the right format, created directly in Matlab.

Chapter 7 describes the input files. We offer three example applications (Mount St. Helens, US - Toba, Indonesia - Vrancea, Romania). In theory, these are the only files the users need to edit once data files are in the correct format.

Chapter 8 describes the output structures and files. The chapter describes all outputs, where you find them and what they represent.

Chapter 9 describes the MuRAT tests. This is a short description of the tests used to assess results, especially of its figure outputs, including those to assess damping.

Chapter 10 describes rays and kernels outputs. The figure outputs show rays and kernels used for each measurement and discuss the transformations from geographic to cartesian.

Chapter 2

Direct-wave attenuation: Q

Timeline in papers.		
Paper	**Descriptor**	**Where**
(Aki and Chouet, 1975)	S-wave CN method	Link
(Yoshimoto et al., 1993)	P-wave CN method	Link
(Del Pezzo et al., 2006)	S-wave CN tomography	Link
(De Siena et al., 2010)	Benchmark with t^* method	Link
(De Siena et al., 2014a)	MuRAT1.0 for CN method published	Link
(De Siena et al., 2014b)	P-wave coda-normalisation tomography	Link
(Prudencio et al., 2015)	Active CN tomography	Link
(De Siena et al., 2017b)	CN tomography measuring Qc	Link

2.1 Parameters

Input Parameters for MuRAT		
Symbol	**Descriptor**	**Where**
t_P or t_S (s)	P-wave or S-wave arrivals	SAC Haeder
t_l (s)	Length of P- or S-wave window	Set in input file
t_0 (s)	Origin time	SAC Haeder
t_c (s)	Lapse time from origin time	Set in input file
t_w (s)	Length of coda window	Set in input file

2.2 The coda normalisation method in pills

The method measures direct and coda-wave energies and divides them to obtain a quantity that only depends on Q, allowing for a linearised inversion.

Pros of measuring direct-wave attenuation using coda-normalised energies

1. Single-station measurements - you can get all the info from a single seismogram.
2. No need for correcting source and site effects.

Cons of measuring direct-wave attenuation using coda-normalised energies

1. Uncertain sensitivity of coda waves to Earth structures - homogeneous coda wave sensitivity assumption is generally unfulfilled. *Tackled by MuRAT estimating the measurable coda-wave energy at the chosen lapse time and assuming a diffusive behaviour.*

2. Correction of radiation pattern - increasing direct wave windows smooth it at the expense of focusing. *Tackled by working with different windows to measure direct-wave energy.*

Figure 2.1 is a visual representation of the method.

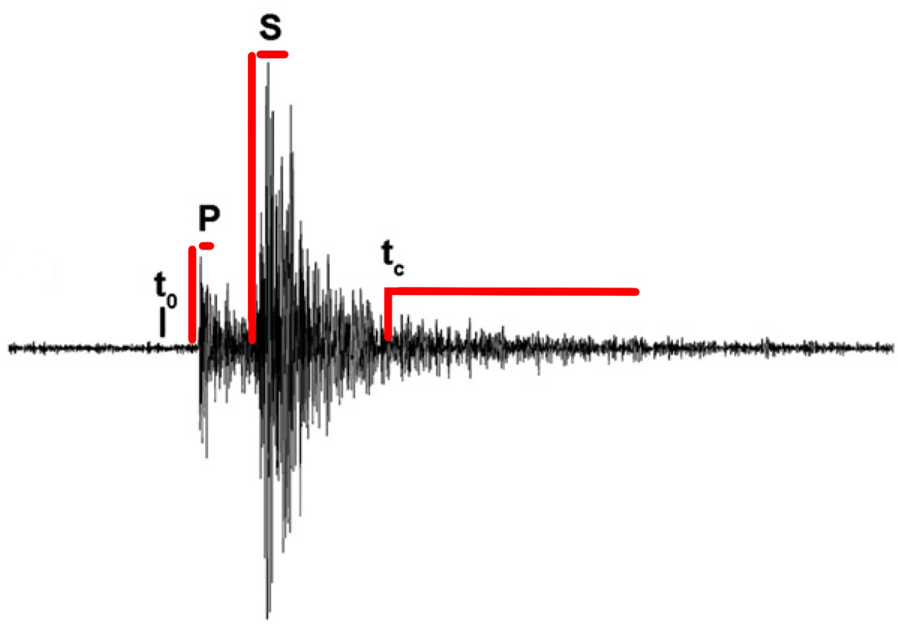


Figure 2.1: The CN method measures the spectral energies of direct and coda waves and inverts for Q. The P and S wave energies, and the normalising coda energies are in red.

2.3 The theory bit

The coda-normalisation (CN) method normalises P- and S-wave energy for coda energy (Aki, 1980; Yoshimoto et al., 1993). Seismic wave energies can be modelled as the convolution of the source, path, site, and instrument functions so that the spectral energy of direct waves (E) is:

$$E(f,r) = R_{\theta\phi} S_S(f) r^{-\gamma} I(f) G(f,\Psi) \exp\left(-\frac{2\pi f}{Q(f)v} r\right) \tag{2.1}$$

where $\boldsymbol{f}$ is the frequency, r is the source-receiver distance, t_c is the coda-wave central lapse time from the origin time of the event (t_0), $R_{\theta\phi}$ is the source radiation pattern (θ and ϕ the azimuth and take-off angle for a source-receiver ray), $S_S(f)$ is the source function, γ the geometrical spreading exponent, $I(f)$ the known instrumental response, $G(f,\Psi)$ is the site amplification factor (with Ψ being the incident angle of the ray at the station), Q is the direct-wave quality factor and v the average velocity in the medium (Yoshimoto et al., 1993).

The spectral energy of coda waves (E_c) can be written as:

$$E_c(f,t_c) = S_S(f)P(f,t_c)G(f)I(f). \tag{2.2}$$

The attenuation of coda waves can then be expressed as (Aki and Chouet, 1975):

$$P(f,t_c) \simeq t_c^{-n}\exp\left(-2\pi f Q_c^{-1} t_c\right) \tag{2.3}$$

In this equation, n is the envelope spectral decay, Q_c is the coda quality factor, and $A_c(f,t_c)$ does not include the effect of the source radiation pattern. We will discuss more about this term in Chapter 4.

Eq. 2.1 can be divided by Eq. 2.2 to normalise the energy of the spectral energy of direct waves using the spectral energy of coda waves:

$$\frac{E(f,r)}{R_{\theta\phi}E_c(f,t_c)} = r^{-\gamma}\frac{G(f,\Psi)}{G(f)}\exp\left(-\frac{2\pi f}{Q(f)v}r\right)\frac{1}{P(f,t_c)} \tag{2.4}$$

where the source function $S_S(f)$ and the instrumental response $I(f)$ contributions disappear. By measuring direct-wave energy over a window of sufficient length (t_l), we smooth the azimuthal contribution of the radiation pattern (De Siena et al., 2009, 2010). If the length of the direct-wave window is chosen appropriately, the contribution of the source radiation pattern $R_{\theta\phi}$ is negligible (De Siena et al., 2009) so that $R_{\theta\phi} = 1$ and $\frac{G(f,\Psi)}{G(f)} = 1$. Since the early coda consists of randomly scattered waves, the coherency and the source radiation pattern are eventually lost (Takemura et al., 2009). De Siena et al. (2010) demonstrate that a coda window of 2 s is sufficient to make the radiation pattern quasi-isotropic in a volcanic caldera. Still, the window length must be carefully chosen at each frequency to avoid the near-receiver onset of surface waves (Gabrielli et al., 2020).

At fixed frequency band and starting lapse-time for coda windows, $P(f,t_c)$ is assumed to be constant in the standard coda-normalisation method (Del Pezzo et al., 2006; Sato et al., 2012). Taking the logarithm, Eq. 2.4 becomes:

$$\ln\left[\frac{E(f,t_c)}{E_c(f,t_c)}\right] = -\ln P(f,t_c) - \gamma\ln(r) - \frac{2\pi f}{Q(f)v}r, \tag{2.5}$$

a linear equation solved for three unknown ($\ln P(f,t_c)$, γ, Q) once the hypocentral distance is known and the frequency is set. If Q_c is known for each source station pair, one can pre-estimate $P(f,t_c)$ using equation 2.3, as done by De Siena et al. (2017b):

$$\frac{1}{2\pi f}\ln\frac{E(f,t_c))}{E_c(f,t_c)} + \frac{1}{2\pi f}\ln P(f,t_c) = -\frac{\gamma\ln(r)}{2\pi f} - Q(f)^{-1}t, \tag{2.6}$$

where t is the travel time of the corresponding direct phase. The system can be solved with a linear inversion for γ and $Q(f)^{-1}$. This is the method implemented in MuRAT3.0.

2.3.1 Applicability

For the applicability of the CN method, one must measure direct-wave energy. This seems simple enough, but it can become problematic depending on frequency, heterogeneity and source-station distance. To check the validity, one must always look at the relationship between direct-wave energy and epicentral distance. Is there a dependence? If not, near field or strong scattering might be prevalent. Does the relationship follow standard geometrical spreading, like those for direct wave energy? How important is the surface wave component? A positive is that generally, where you cannot measure *direct-wave energy*, you can use codas, i.e., diffusion onsets pretty quickly (see chapter 4).

2.4 MuRAT workflow

MuRAT works in analogy to De Siena et al. (2010), De Siena et al. (2014a) and Sketsiou et al. (2021) when mapping peak delays. These authors show that one must check the relationship between hypocentral distance (travel time) and measured energy ratio. In Figures 2.2, we show an example at Mount St. Helens volcanoes, where sources and stations can be either on or inside the volcanic edifice and near surroundings (red dots) or far from them (cyan dots). The expected energy decrease with hypocentral distance is unfulfilled at small distances, where surface and trapped waves dramatically increase coda energy.

The energy in the coda can be back-traced to mark the position and intensity of the corresponding

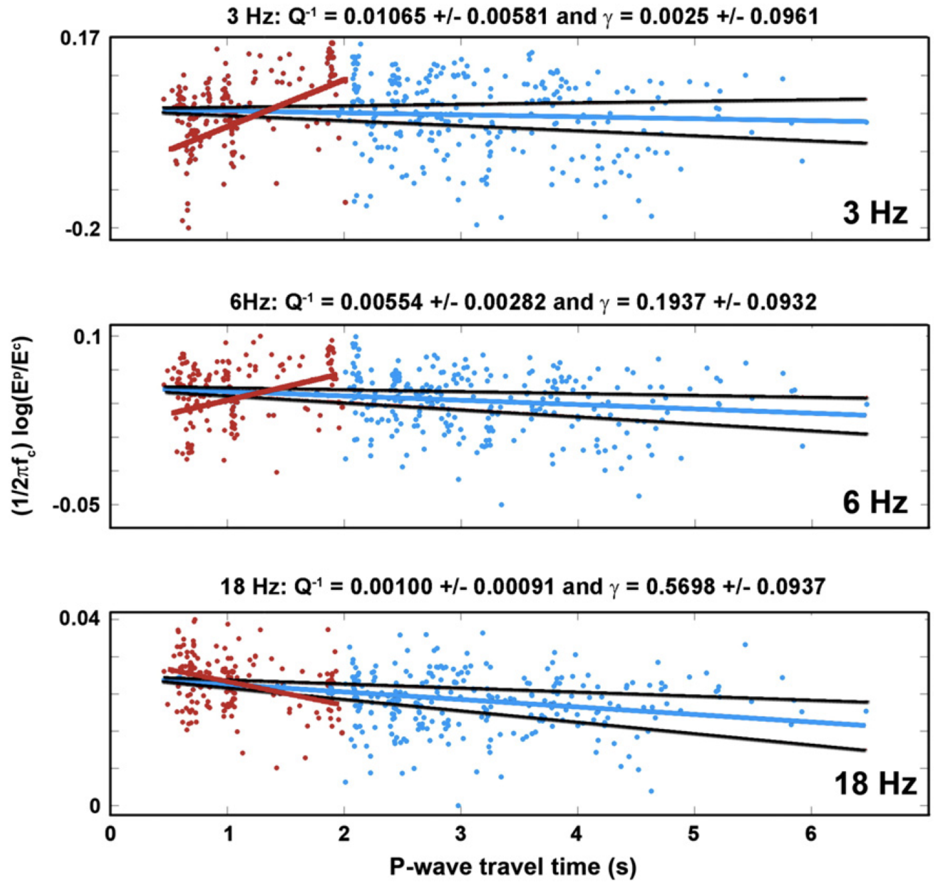


Figure 2.2: A plot of the logarithm of the P-wave direct-to-coda energy ratios (red and cyan dots, computed at MSH in three frequency bands) versus the P-wave travel times at Mount St. Helens volcano. The red lines show the linear fit obtained using only data between 0 and 2 s travel times, affected by surface waves inside the volcanic edifice. The cyan lines represent the least squares fit for the entire dataset. The black lines are the maximum uncertainties. The geometrical spreading and the average Q obtained from the inversion are shown above each plot. Notice the progressive change with increasing frequency. From De Siena et al. (2014a).

scatterers (De Siena et al., 2014b), using scattering tomography (also known as Nishigami (1991) technique). The method can still be applied, but fixing the geometrical spreading will trade-off measurements at low frequencies. Sketsiou et al. (2020) discusses the corresponding trade-offs in volcanic, fault and regional settings. In Fig. 2.3, we show the same plot for the normalised S-wave energy at Mount Vesuvius, where all sources and stations are inside the more heterogeneous volcanic edifice and feeding system domain. Here, only data at 18 Hz can be used. Following De Siena et al. (2014a), measurements above (below) the logarithmic, linear trend show high (low) attenuation. This procedure is inverted by this version of MuRAT using a ray bending approach (Block, 1991; De Siena et al., 2010), discussed computationally in Chapter ???.

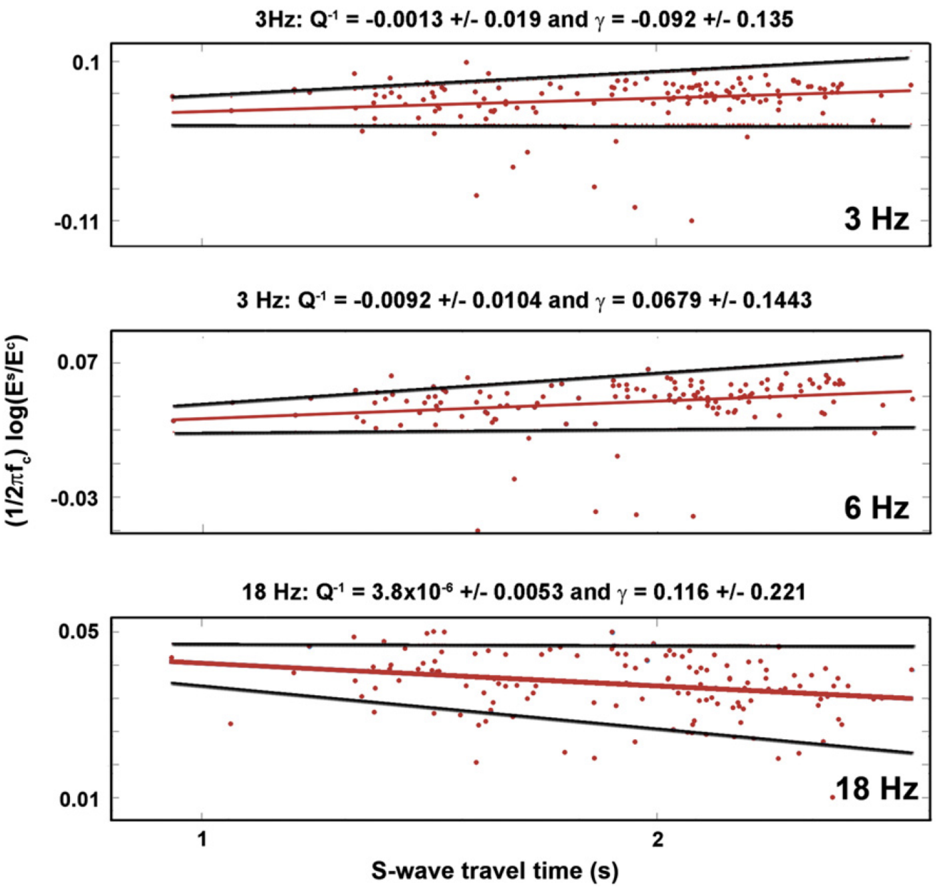


Figure 2.3: Same as Fig. 2.2 for Mount Vesuvius, where earthquakes and stations are all comprised under the volcanic edifice. Notice the negative Q at low frequencies. From De Siena et al. (2014a).

Chapter 3

Scattering attenuation: *peak delays*

Timeline in papers.		
Paper	**Descriptor**	**Where**
(Saito et al., 2002)	Markov approximation	Link
(Takahashi et al., 2007)	Peak-delay regionalisation	Link
(Takahashi et al., 2009)	Peak-delay tomography	Link
(Calvet et al., 2013)	Peak delay regionalisation + Qc: crust	Link
(De Siena et al., 2016)	Peak delay + Qc: volcano	Link
(Sato, 2016)	Peak delays - short and long λ	Link

3.1 Parameters

Input Parameters for MuRAT		
Symbol	**Descriptor**	**Where**
t_{min}	minimum peak delay allowed	Set in input file
t_{max}	maximum peak delay allowed	Set in input file
t_0 (s)	Origin time	SAC Haeder

3.2 The peak delay method in pills

The method measures the delay of the maximum energy of the envelope of the direct wave to model scattering attenuation with regionalisation.

Pros of measuring scattering attenuation using peak delay

1. Single-station measurements - you can get all the info from a single seismogram.
2. Proven sensitivity to structural information and geologic boundaries.

Cons of measuring scattering attenuation using peak delay

1. Assumptions underlying the Markov approximation could be unfulfilled. *Tackled by MuRAT estimating parameters at different frequencies.*
2. No inversion, only regionalisation. *Still not tackled by MuRAT.*

Figure 3.1 is a visual representation of the method.

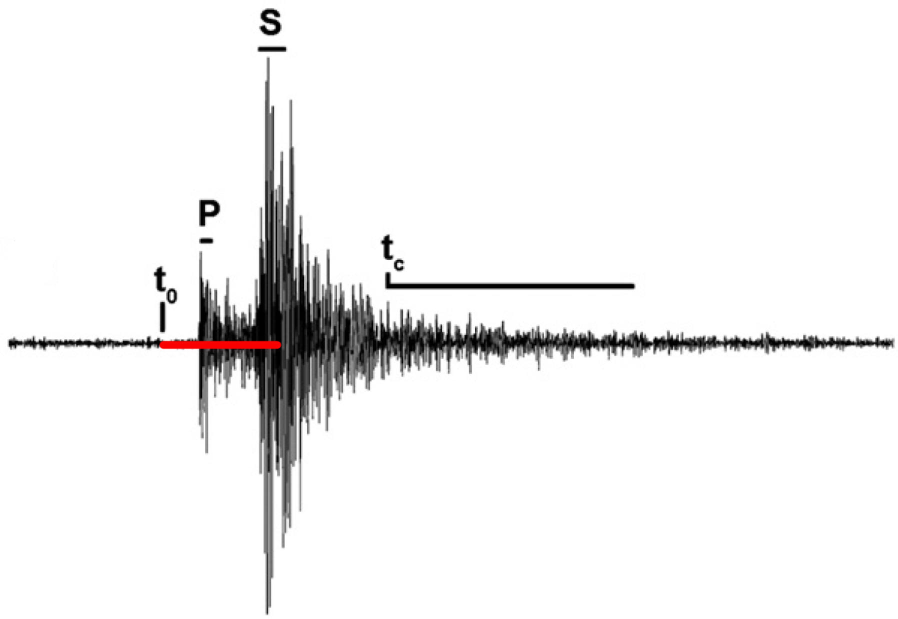


Figure 3.1: The PD method measures the delay of the envelope's maximum from either the arrival time or the origin time of the event.

3.3 The theory bit

While travelling, direct wave packets broaden because of scattering and diffraction. The seminal work for the application of *peak delay* (envelope broadening) imaging is Aki and Chouet (1975), which demonstrates that the coda of high-frequency seismograms originates from random inhomogeneity in elastic properties, and (Sato, 1989), whose first proposed peak-delays as a tool to characterise random heterogeneity. Fehler et al. (2000) proved that the Markov approximation for the parabolic wave equation can synthesise seismic waveforms in two dimensions, including early and intermediate coda. The seminal work for the technique is Saito et al. (2002), who developed the equations for forward modelling envelopes for spherically outgoing media characterised by von Kármán power spectral density functions (PSDF).

Saito et al. (2002) provides both a historical view and a discussion of the assumptions underlying the method:

- The primary assumption of the method is that the wavelength ($\lambda = 2\pi/k$) is smaller than the correlation distance (a), so if this is multiplied by the wavenumber: $ak >> 1$. The estimation of a is challenging, but it appears likely that the assumption will be broken at low frequencies, where *diffraction* takes over.

- Propagation is entirely constrained within a single parallelepiped of average velocity V_0. The velocity field depends on space (r) as:

$$V(r) = V0(1 + \xi(r)) \tag{3.1}$$

  where $\xi(r) << 1$ are random velocity fluctuations. For a wider description of how to derive auto-correlation and power spectral density functions describing fluctuation distributions, refer to Sato et al. (2012). These quantities are controlled by ϵ (the root-mean-square of velocity fluctuations), a (the correlation length) and κ (tuning the richness of short-wavelength components of the random medium).

- Frequencies are above 1 Hz; below that, the wavelength grows to a level where most assumptions are invalid in lithospheric settings.

The original definition models envelope duration (t_q) as the quantity related to the characteristic time of envelope broadening (t_M) (Saito et al., 2002). It is instructive to look at the definition of t_M given by Sato (2016):

$$t_M(\kappa, \zeta, k_c, r_0) = \frac{\epsilon^2}{2V_0 a} C_L(\kappa, \zeta, k_c) r_0^2; \tag{3.2}$$

where ζ is a number varying between 0.25 and 1.75. Sato (2016) defines it as a tuning parameter and separates random media between long- and short-scale components so that the correlation distance for the small-scale component is: $a_S^{-1} = \zeta k_c$.

This quantity fully defines scattering in the case of long wavelength heterogeneities, depending on average velocity (V_0), hypocentral distance (r_0, which in a homogeneous medium is the product of V_0 and travel time of the phase t), correlation distance (a), and mean squared velocity fluctuations (ϵ^2), the wavenumber, depending on the angular frequency (ω_c) as $kc = \frac{\omega_c}{V_0}$ and where:

$$C_L(\kappa, \zeta, k_c) = \begin{cases} \frac{\pi^{1/2}\Gamma(\kappa+\frac{1}{2})}{(2\kappa-1)\Gamma(\kappa)}(1 - (\zeta a k_c)^{1-2\kappa}) & \kappa \neq \frac{1}{2} \\ 2ln(\zeta a k_c) & \kappa = \frac{1}{2} \end{cases} \tag{3.3}$$

The dependency on V_0 and a seem straightforward until we do not make r_0 and $C_L(\kappa, \zeta, k_c)$ explicit. Let's see what happens for $\kappa = \frac{1}{2}$:

$$t_M(\kappa = \frac{1}{2}, \zeta, k_c, r_0) = \frac{\epsilon^2}{2V_0 a} V_0^2 t^2 2ln(\zeta a k_c) = \frac{\epsilon^2 V_0 t^2 ln(\zeta a k_c)}{a}; \quad (3.4)$$

in which case:

- The characteristic time increases quadratically with travel time, hence the need for correcting this dependency to focus on scattering.
- A sure way to increase scattering is to increase ϵ^2 -beware that these variations are generally limited.
- Characteristic times increase with frequency and ζ although only logarithmically.
- For actual ranges of correlation distances, t_M can change a lot.

3.3.1 Applicability

The important assumption for the applicability of peak-delay analysis is that long-scale components produce scattering. To check the validity, one must always test the following conditions: $\zeta a k_c >> 1$. In general, this means a frequency above 1 Hz. In this case, we can rewrite:

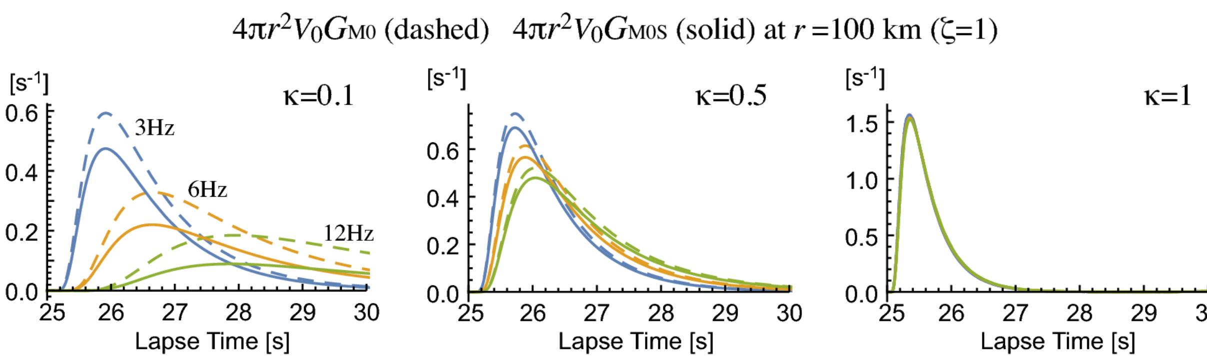


Figure 3.2: Broadening for different κ values. For parameters used, see Sato (2016).

$$t_M(\kappa,\zeta,k_c,r) = \begin{cases} \frac{\epsilon^2}{2V_0 a} r^2 \frac{\pi^{\frac{1}{2}}\Gamma(\kappa+\frac{1}{2})}{(1-2\kappa)\Gamma(\kappa)} (\zeta a k_c)^{1-2\kappa} & \kappa < \frac{1}{2} \\ \frac{\epsilon^2}{2V_0 a} r^2 ln(\zeta a k_c) & \kappa = \frac{1}{2} \\ \frac{\epsilon^2}{2V_0 a} r^2 \frac{\pi^{\frac{1}{2}}\Gamma(\kappa+\frac{1}{2})}{(2\kappa-1)\Gamma(\kappa)} & \kappa > \frac{1}{2} \end{cases} \tag{3.5}$$

This is probably the most important relationship as it shows the lack of changes with frequency for $\kappa > \frac{1}{2}$ and the slow dependency for $\kappa = \frac{1}{2}$. Characteristic times, envelopes and scattering are strongly frequency-dependent when $\kappa < \frac{1}{2}$. This gives us a quick test of our data. Do waveforms broaden dramatically at high frequencies (Fig. 3.2)? This is already an indication of the richness in long wave numbers (short-scale components). The first check the user should do is then visualise the data.

3.4 MuRAT workflow

MuRAT works in analogy to Takahashi et al. (2007), Tripathi et al. (2010) and Calvet et al. (2013) when mapping peak delays. These authors show that above 1 Hz, one must check the relationship between hypocentral distance and measured peak delay. We offer two examples of different hypocentral distances in Figures 3.3 and 3.4. There is a threshold over which you can apply the approximation. Hence Murat shows you the same plot for your dataset - we will discuss in Chapter ?? the examples at Mount St. Helens, Vrancea and Toba. The peak-delay method requires the same ray tracing approach used for the CN method, discussed computationally in Chapter **??**.

Following Takahashi et al. (2007), measurements above (below) the logarithmic, linear trend shows high (low) scattering. This version of MuRAT regionalises this procedure. It means positive and negative variations are allocated via their average over blocks after raytracing. The hit count of each segment in each block weights the variation. Takahashi et al. (2009) show that there are better procedures to invert for the characteristics of power spectral density function using peak delay: we leave it for future versions of the code.

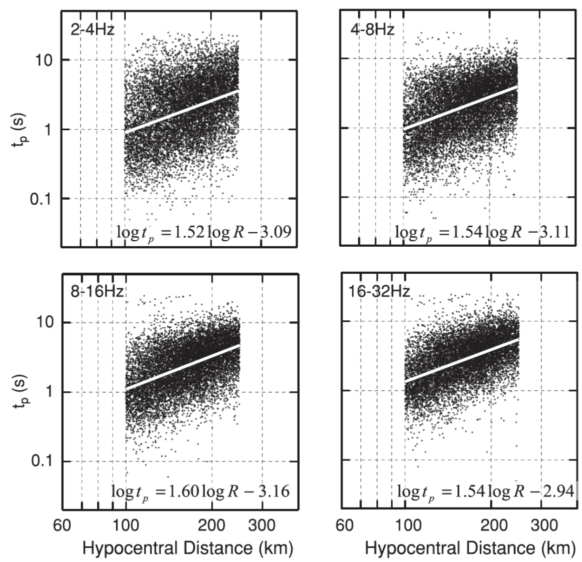


Figure 3.3: Logarithmic plot of peak delay times against hypocentral distances. Black dots represent the data used in this study. White lines are regression lines. From Takahashi et al. (2007).

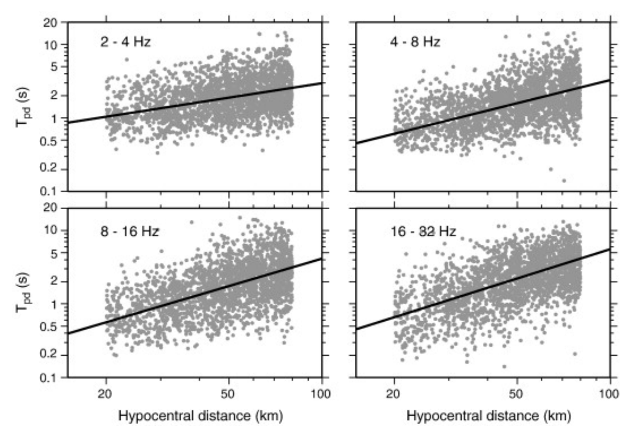


Figure 3.4: Logarithmic plot of peak delay times (in seconds) as a function of the hypocentral distance (in kilometres) for crustal S waves in four frequency bands. Grey dots are the data, and black lines are the regression lines: log10(Tpd) = Ar(f) + Br(f)log10R. From Calvet et al. (2013).

Chapter 4

Absorption: *coda attenuation*

Timeline in papers.		
Paper	**Descriptor**	**Where**
(Aki and Chouet, 1975)	Origin of Coda	Link
(Fehler et al., 1992)	MLTWA	Link
(Calvet et al., 2013)	Peak delay regionalisation + Qc: crust	Link
(De Siena et al., 2016)	Peak delay + Qc: volcano	Link
(Del Pezzo et al., 2016)	2D RTT sensitivity kernels: analytic	Link
(De Siena et al., 2017a)	2D inversion with analytic kernels	Link
(Del Pezzo et al., 2018)	3D RTT sensitivity kernels: diffusive	Link
(Akande et al., 2019)	3D inversion with diffusive kernels	Link
(Sketsiou et al., 2020)	Absorption tomography	Link

4.1 Parameters

Input Parameters for MuRAT		
Symbol	**Descriptor**	**Where**
t_k	starting lapse time	Set in input file
t_w	coda window length	Set in input file
t_0 (s)	Origin time	SAC Haeder

4.2 The Qc method in pills

The method measures the envelope's decay from a given lapse time to model absorption with an inversion procedure.

Pros of measuring absorption using Qc

1. Single-station measurements - you can get all the info from a single seismogram.
2. Proven sensitivity to fluids.
3. High-resolution information on near-source and near-station structures.
4. Full inversion procedure.

Cons of measuring scattering attenuation using peak delay

1. The diffusive assumption is rarely fulfilled, with the onset of surface waves corrupting especially low frequencies. *Tackled by MuRAT estimating parameters at different frequencies.*

2. Assumption of single highly-diffusive layer rarely fulfilled. *Still not tackled by MuRAT.*

Figure 4.1 visually represents the method.

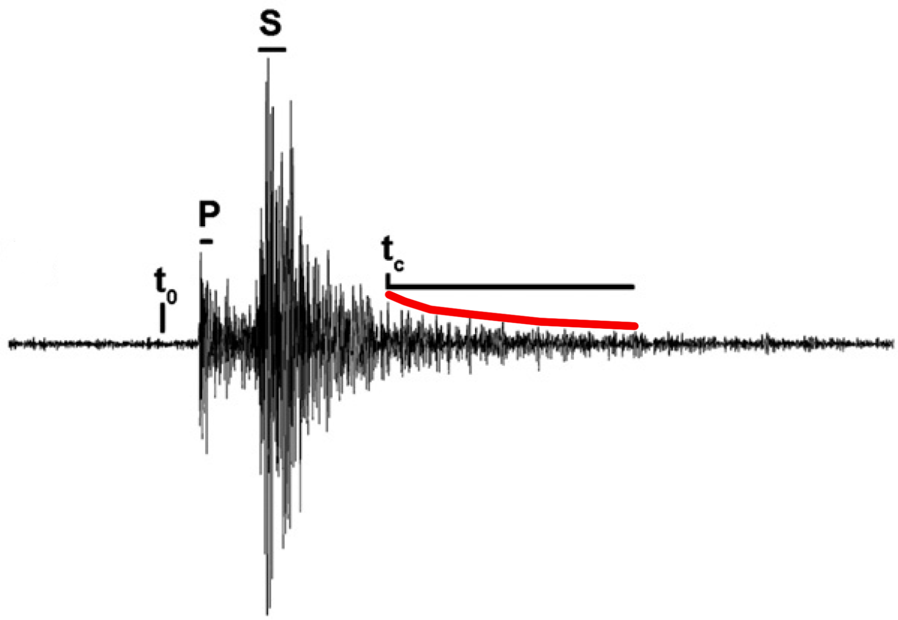


Figure 4.1: The Qc method measures the envelope's decay from a given lapse time from the origin time of the event.

4.3 The theory bit

4.3.1 MLTWA

Coda waves can be used for more than just normalising direct wave energies. Starting from the seminal work of Aki and Chouet (1975), coda waves have been used to characterise the tectonic state of the Earth. This has led to the development of the Multiple Lapse Time Window Analysis (MLTWA) technique Fehler et al. (1992) to separate scattering attenuation Q_s^{-1} from absorption Q_i^{-1}. It does so by measuring extinction length ($L_e = Q_s^{-1} + Q_i^{-1}$) and seismic albedo ($B_0 = \frac{Q_s^{-1}}{L_e}$).

Two assumptions are necessary to apply the MLTWA technique: the direct S-wave dominates the early portion of an S-wave seismogram; the S-coda comprises scattered S-waves. While the total attenuation of the medium controls the first S-wave amplitude, coda-waves are a product of scattering (Sato et al., 2012). The filtered seismograms are divided into three time-windows, starting from the S arrival, defined by a starting time t_k and window length t_w. The time integrals $E_k(r)$ of the seismic energy density $E(r,t)$ in each window k is:

$$E_k(r) = \int_{t_k}^{t_k+\Delta t} E(r,t)dt \quad k = 1...3 \tag{4.1}$$

where the energy is still a function of the source-station distance *r*. The wave energy decay for each cycle $-\Delta E/E$ is a function of the total quality factor Q (related to the attenuation coefficient η) via the following equation:

$$-\frac{\Delta E}{E} = \frac{2\pi}{Q} = 2\pi\left(\frac{1}{Q_i} + \frac{1}{Q_s}\right) = \frac{\upsilon\eta}{f} \tag{4.2}$$

where υ is the average wave speed and f is the frequency, respectively. The next step consists of correcting the energy of each window $E_k(r)$ for both the geometrical spreading and the integral of the t_w-seconds-long normalisation window (the last window in time). This procedure removes source and site effects (Mayeda et al., 1992):

$$E_k^{obs}(r) = \log_{10}\left(4\pi r^2 \frac{E_k(r)}{\int_{t_{coda}}^{t_{coda}+\Delta t} E_k(r,t)dt}\right) \tag{4.3}$$

The resulting logarithms are plotted versus distance in the four frequency bands considered and fitted to the theoretical normalised energies. The theoretical curves are modelled using the L2-norm misfit function for several waveforms from i to N:

$$M(L_e^{-1}, B_0) = \sum_{i=1}^{N}\sum_{k=1}^{3}[E_k^{obs}(r_i) - E_k^{theo}(r_i, L_e^{-1}, B_0)]^2 \tag{4.4}$$

$E_k^{theo}(r_i, L_e^{-1}, B_0)$ is the theoretical normalised energy computed at distance r_i for the k-th time-window, fixing the L_e^{-1} and B_0 obtained from MLTWA. M is minimised with a grid search in the B_0 and L_e^{-1} parameter space, and the best-fit values $\hat{L}_e^{-1}$ and $\hat{B}_0$ are given by the minima of the function. The isolines of the variable $M_{norm} = M(L_e^{-1}, B_0)/M(\hat{L}_e^{-1}, \hat{B}_0)$ give the error estimates on $\hat{L}_e^{-1}$ and $\hat{B}_0$. Being M_{norm} an F-variable (Del Pezzo and Bianco, 2010), the model parameters can be estimated using an F distribution (Pisconti et al., 2015), with a level of confidence higher than F=0.68.

4.3.2 Sensitivity kernels

Due to the increased lateral sensitivity compared to ray-dependent techniques, coda waves are theoretically the ideal tool to image the Earth. In practice, apart from the cases of single scattering and diffusion, there is generally no simple analytical solution, and envelopes have to be dealt with Radiative Transfer Theory (Sato et al., 2012), possibly mixed with ray-tracing techniques that model direct wave information (Sanborn et al., 2017). However, if the wave field is diffusive, one can obtain coda sensitivity kernels with Paasschens equations (Paasschens, 1997) and the approximations of Pacheco and Snieder (2005). Several authors have obtained 2D diffusive kernels based on Radiative Transfer Theory (Obermann et al., 2013; Margerin, 2013; Mayor et al., 2014; Del Pezzo et al., 2016). These kernels have maximum sensitivity at source and receiver (they show a pole at their exact location that must be interpolated) and are all very similar (see, e.g. Figs. 4.2) despite being obtained with different approximations. They primarily aim at coda-wave interferometry but have found applications in coda-attenuation imaging at regional and local scales (Mayor et al., 2016; De Siena et al., 2017a; Napolitano et al., 2020; Sketsiou et al., 2020). (Calvet et al., 2013) first noticed the contribution of surface waves to low-frequency observations, an effect modelled in 2D by Obermann et al. (2013) and fit on data by Gabrielli et al. (2020). Del Pezzo et al. (2018) have extended the formulation to 3D media, setting the spatial and frequency limits of the kernel applicability and discussing the assumptions behind the applicability of the kernels.

There are two published approaches to map coda attenuation in the 3D space with kernels. The first is regionalisation (Del Pezzo et al., 2018) in analogy to peak-delay mapping. An excellent review paper on this approach is Del Pezzo and Ibáñez (2020). MuRAT follows instead the pathway laid down by De Siena et al. (2017a). It assumes that all coda energy is contained within the propagation grid at the chosen lapse time. If true, coda attenuation in space can be obtained from source-station measurement with a standard inversion. Sketsiou et al. (2020) extensively discusses the differences between regionalisation and inversion. Akande et al. (2019) is the first 3D kernel-based coda attenuation model obtained with 3D sensitivity kernels and an inversion approach.

As anticipated in Chapter 1, the combined effect of coherent and incoherent scattering as well as the trade-off induced by changing scattering regimes (from Rayleigh to Mie scattering, Cormier and Sanborn (eg 2019)) can profoundly affect the reliability of the kernels. For these to be applied, there is

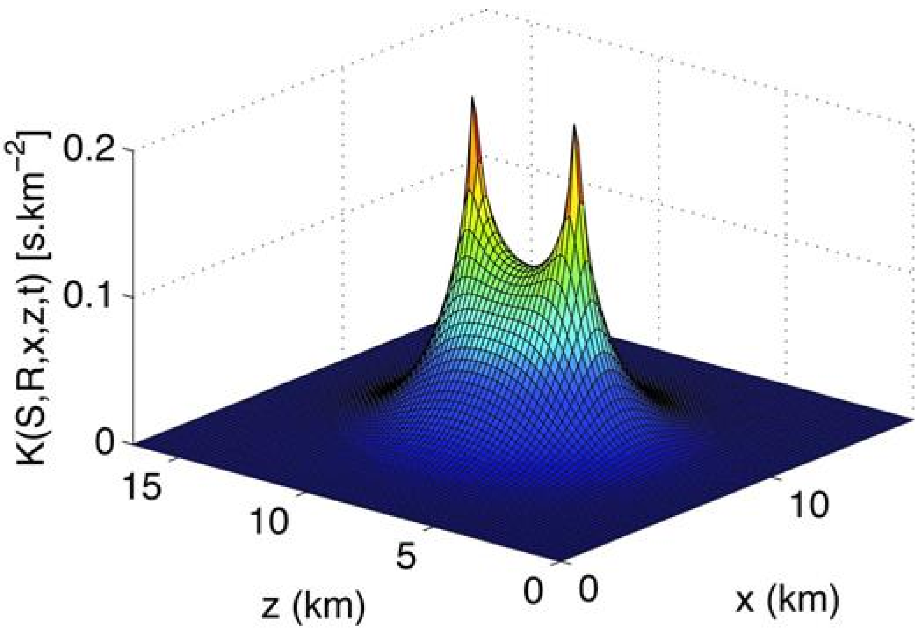


Figure 4.2: Diffusive sensitivity kernels at time 3.6 s obtained by Obermann et al. (2013)

a need for a volume with uniform (or gradually changing) scattering properties, which is rarely valid at all frequencies investigated. Surface Obermann et al. (2013) and resonant Margerin (2013); De Siena et al. (2013) waves, sharp horizontal transitions in scattering (Wegler, 2005; Margerin, 2017; Nardoni et al., 2021) and more generally not accounting correctly for the trade-offs with geometrical spreading (Morozov, 2011) all bring to inexact, if not incorrect, images of the Earth's scattering and absorption properties.

4.3.3 Applicability

MuRAT measures coda attenuation in the diffusive assumption; hence one must check that Q_c^{-1} does not vary with hypocentral distance (Calvet et al., 2013). However, this only attests that equipartition is in place, a necessary but insufficient condition for diffusion. The analysis of coda attenuation with frequency is central to testing the applicability of both MLTWA and sensitivity kernels (Sketsiou et al., 2020). Also, as we are using the inversion procedure, it is paramount to test that most coda energy remains inside the inversion grid (Akande et al., 2019).

MuRAT offers tools to test these approximations and the effect of Qc on measurements of total attenuation (chapter 2).

4.3.4 MuRAT workflow

MuRAT works in analogy to Mayor et al. (2016), De Siena et al. (2017a) and Akande et al. (2019) when modelling coda attenuation. These authors show that above 1 Hz, one must check the relationship between hypocentral distance and measured coda attenuation. In Figure 4.3, we show the expected relation of coda attenuation versus hypocentral distance for the theory's applicability and interpretation of Qc as an absorption measurement. It is clear that the standard assumption of twice the S-wave travel time (Aki and Chouet, 1975) does not work. Hence Murat shows you the same plot for your dataset. However, the ideal threshold (100 s) removes too many data for reliable coverage. Figures 4.3c shows the acceptable compromise.

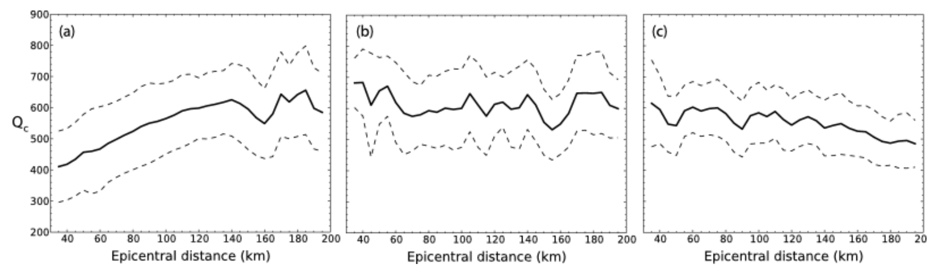


Figure 4.3: (a) Qc as a function of the epicentral distance in the 4–8 Hz frequency band. The duration of the coda window is fixed (t_w = 50 s). The solid and dashed lines show the mean dependence of Qc with the epicentral distance and the associated uncertainties, respectively. The analysis is performed on the onset of coda window t_k equal to (a) 2tS, (b) 100 s, and (c) 70 s. From Mayor et al. (2016)

MuRAT also outputs the sensitivity kernels in two grids: (1) the fine propagation grid, extending generally well beyond the area of inversion, and (2) the inversion grid. The first is to compute the kernels and interpolate the poles at source and receiver locations (Akande et al., 2019). The user chooses the boundaries of the second depending on source-station coverage. The code shows what percentage of the energy is still inside the inversion grid.

Chapter 5

Getting started

5.1 Installation and set up in a nutshell

The current version of the code works following these steps:

1. Download or clone the package at the corresponding GitHub page;

2. Work in the downloaded folder after moving it to an appropriate location on your system;

3. Check that the IRTools have been downloaded in the corresponding folder in the working directory. Otherwise download them from the corresponding GitHub page;

4. Open one of the three input .mlx files, providing a step-by-step explanation of all inputs ($Murat\_inputMSH.mlx$, $Murat\_inputRomania.mlx$, or $Murat\_inputToba.mlx$) and create your own;

5. MuRAT works with SAC files that must be stored in a single folder and corrected for the instrument function. The files must have populated headers, although the code can work using just some header fields (see Chapter 6);

6. Run MuRAT3.mlx and select the name of the input file desired.

5.2 Matlab

MuRAT wants to be an open-access code, yet it also comes in Matlab format: why?

- The first reason is historical. The code has been designed in Matlab, with legacy bash and Fortran codes embedded in it. Passing from Matlab to Python, the obvious choice for open-access interfaces was not straightforward. Passing from Matlab to Julia has been way easier.

Evolving through a decade, the Matlab version of MuRAT has reached the format presented here. Yet, MuRAT3.0 will be the last release in Matlab.

- The second reason is the simplicity of learning. MuRAT has been thought of as a way to introduce students to computational seismology. Matlab is an ideal interface for teaching computational codes to students in Earth Sciences. MuRAT has also been developed with the wider geological community in mind, which until recently was often lacking training in computation. The average geology student understands Matlab better than other codes.

- The third reason is that Python is becoming the standard in seismology, mainly thanks to tools like Obspy, yet this is not so obvious for the rest of Earth Sciences. For example, Matlab is (has been) the favourite language interface for most of the Geodynamics community - see, for instance, Taras Gerya's Introduction to Numerical Geodynamic Modelling. The geodynamics community is now embracing Julia as its natural successor. MuRAT does the same.

5.2.1 Requirements and installations in pills:

- *DOWNLOAD:* MuRAT from Github (Fig. 5.1). Ideally, fork or star the code to get updates. This manual explains how to work with GitHub Desktop.

- *SYSTEM:* The program works on Mac, Linux and Windows systems equipped with Matlab R2019a or later.

- *NECESSARY TOOLBOXES:* Signal Processing, Curve Fitting, Image Processing and Mapping Toolboxs. The Parallel Computing Toolbox is recommended for speed.

- *EXAMPLE DATASETS:* Three sample datasets (Mount St. Helens, Romania, and Toba) are included in the package, allowing the user to obtain sample models. The datasets work with the input **.mlx** files that are provided in the working directory.

5.2.2 Changing folder

In this example, MuRAT is cloned from the corresponding page (Fig. 5.1) to a folder inside GitHub Desktop. As a developer or user of MuRAT for multiple datasets, you might want to copy this folder on your system at a different location (Fig. 5.2). In the example (Fig. 5.2,a), we show how you automatically download the main version of the code in a Github folder. Depending on your system,

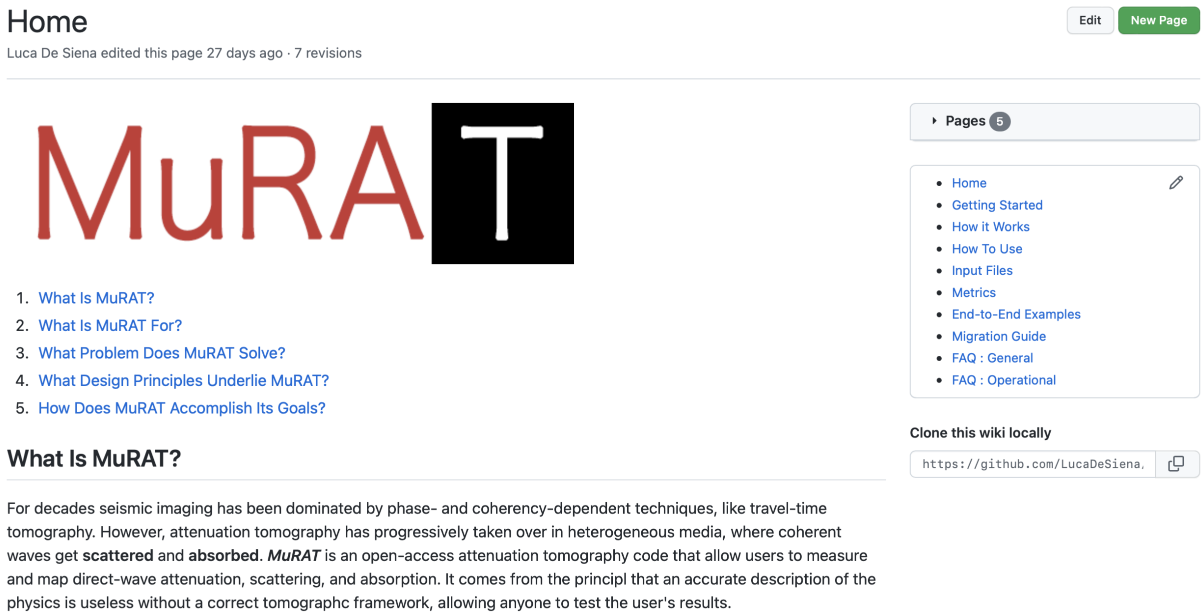


Figure 5.1: The MuRAT page on GitHub.

the path to this folder could be different. It is recommended to copy and rename the folder to another path (Fig. 5.2,b), and leave the original MuRAT folder where it is downloaded. When finished with your work or if cooperating on that specific dataset with someone, you can then substitute your edited folder in Github. This will allow you to push a new branch that might be used as a repository for codes and results. These repositories are now a requirement for most journals and will enable you to get the seminal version of your results in case of reviews.

5.2.3 README.md

The simplest way to start with MuRAT is by reading its readme.txt. This file provides instructions on installing the code and setting up your data, workflow, and outputs. This is the lowest level of documentation but covers the basics without going into jargon or trying to solve all problems at once.

a)

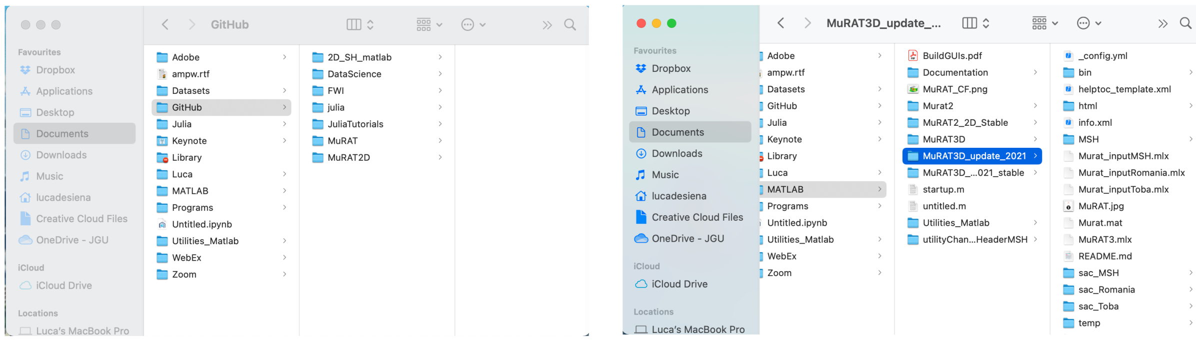


Figure 5.2: How I work with MuRAT in Matlab. a) The folder where the branched code is downloaded (left) vs the working directory (right). b) The Workspace in Matlab. Notice that, in theory, only the Matlab input files are in the folder and must be edited if the data are prepared correctly.

5.2.4 Wiki

Higher-level documentation is provided by the MuRAT Wiki (Fig. ??). Like any wiki on GitHub, it has a lateral panel that allows reaching pages that explain data preparation or input files, for example. The primary reason why the user should use MuRAT is the videos. Several pages present videos that explain (Luca De Siena's) way of working with the code. These videos might be the simplest, most straightforward way to use the code.

5.3 Julia

MuRAT will be an open-access code developed in Julia from this version onward.

Julia is FAST. Julia is EASY. Julia is more similar to Matlab. Hence, the choice to switch from Matlab to Julia. Is it better than Python? This is an absurd question for two languages developed in different decades. If you need to call a Python script (and sometimes you do), you can do it from Julia. Please have a read at this interesting comparison to know more about how the two codes compare.

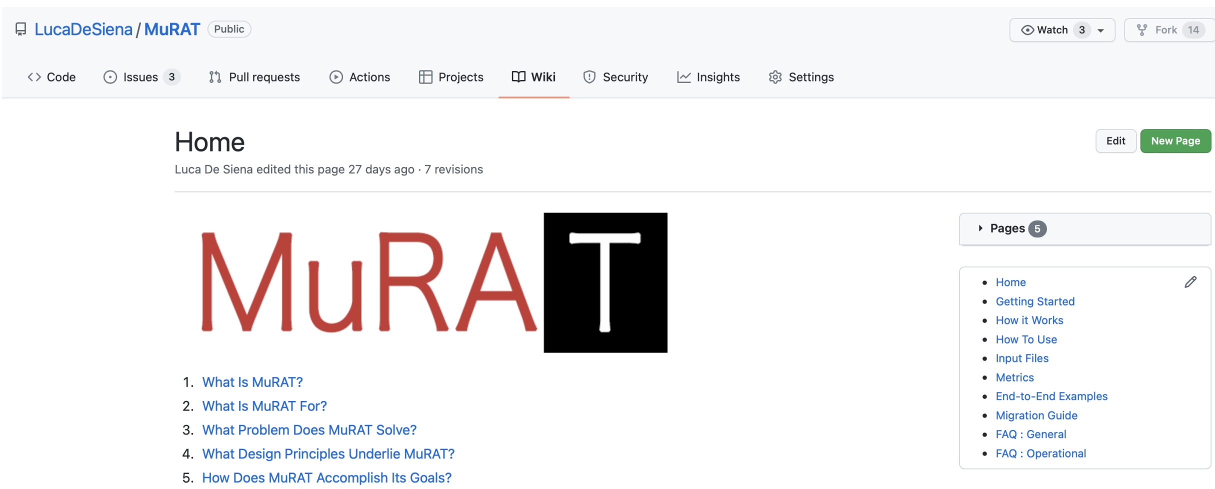


Figure 5.3: MuRAT has a Wikipedia-like environment that provides specific documentation on tasks and structures of the code.

Chapter 6

Data preparation

6.1 SAC files

The Seismic Analysis Code has been the standard in seismology for at least 20 years. To process the SAC data in Matlab, we use the MatSAC tools created by Prof. Zhigang Peng from Georgia Tech. MuRAT owes a lot to the format defined by Peng with his $fget\_sac$ function:

$$[t, data, SAChdr] = fget\_sac(pathSACFile) \tag{6.1}$$

where, given the path of the SAC file, the user obtains two variables (*time* and *data*) and especially a structure (*SAChdr*) that stores the fields of the header.

This chapter deals with the original SAC files and how to populate their header to work with MuRAT. By populating the header, the user will need no additional file related to data processing.

6.2 Parameters

The aim of MuRAT has always been to do attenuation imaging using SAC files that can be downloaded from online servers. These data are provided via miniseed that must be unpacked. The resulting SAC might have already populated headers: it is up to the user to populate the parameters shown in the following table if they are blank (or *-1234* as by SAC standard).

Necessary Parameters in SAC header		
SAC field	**Descriptor**	**Name in Matsac header**
o (s)	origin time	$SAChdr.times.o$
a (s)	P-wave pick	$SAChdr.times.a$
t0 (s)	S-wave pick	$SAChdr.times.t0$
stla (deg)	station latitude	$SAChdr.station.stla$
stlo (deg)	station longitude	$SAChdr.station.stlo$
stel (m)	station elevation	$SAChdr.station.stel$
evla (deg)	event latitude	$SAChdr.event.evla$
evlo (deg)	event longitude	$SAChdr.event.evlo$
evdp (km)	event depth	$SAChdr.event.evdp$

6.3 $sac\_datafolder$ - checking data

The bold-faced parameters in the above table are mandatory. The user can check the parameters in the header of each SAC file using the utilities $Murat\_test.m$ and $Murat\_testAll.m$ in the **Utilities_Matlab** folder. After running MuRAT the first time, this folder will be added to your path. Otherwise, copy these scripts to your working directory folder.

6.3.1 $Murat\_test$ - testing single waveforms

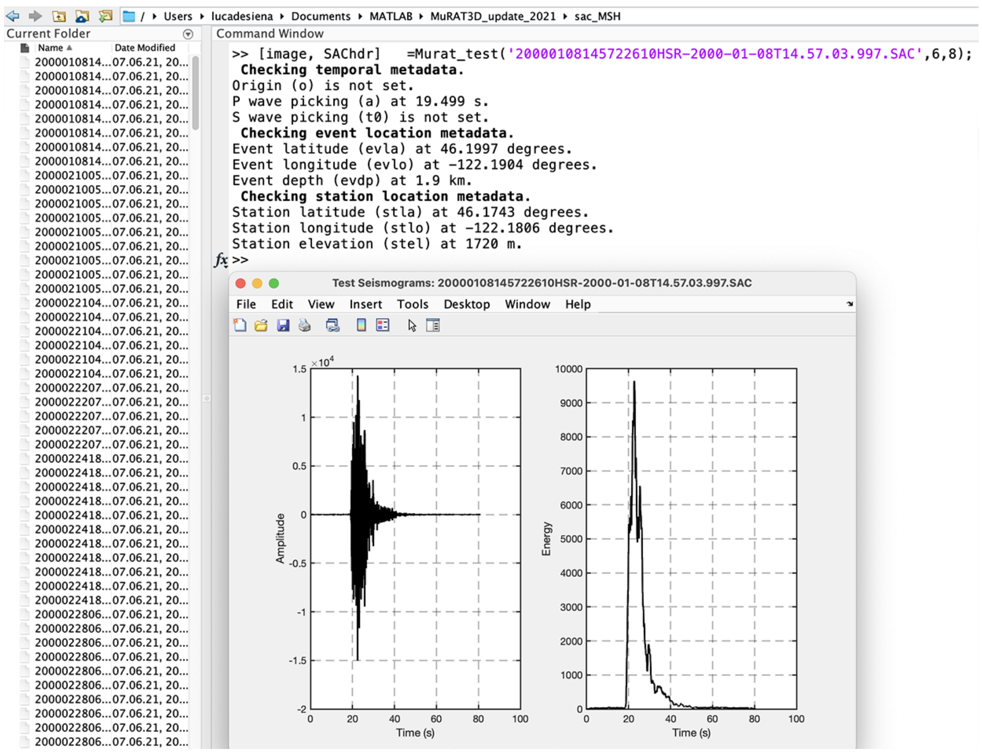


Figure 6.1: Example run of Murat_test. The image is shown while the SACheader is created in the workspace. All the necessary parameters are checked. Notice the missing S-wave picking - the variable storing this parameter may be different from t0.

```
% function [image, SAChdr]    =   Murat_test(nameWaveform,...
%    centralFrequencies,smoothingC,figOutput,verboseOutput)
% TEST seismogram envelopes for changes in  broadening
```

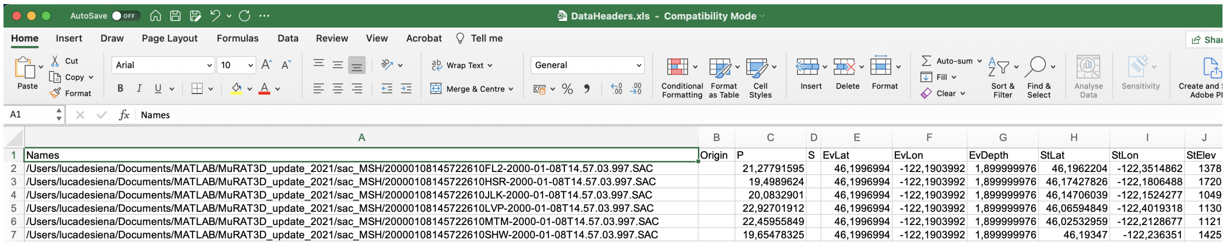
```
% CREATES a figure with seismograms and envelopes for different frequencies
%
% Input Parameters:
%       nameWaveform:           name of the SAC file
%       centralFrequencies:      vector of frequencies (Hz), if [] no filter
%       smoothingCoefficient:   coefficient to smooth envelopes
%       figOutput:                     decide if you want to show figures (set to 1)
%       verboseOutput:             decide if you want to display messages (set to 1)
%
%   Output:
%       image:                          image with the envelope at a specified frequency
%       SAChdr:                         header of the SAC file
%
```

The function uses the path of a waveform, a vector of central frequencies, and a smoothing coefficient as inputs. Two additional flags allow us to show the corresponding seismogram and envelope (set $figOutput = 1$) and the output messages (set $verboseOutput = 1$). It provides the user with the header structure (SAChdr). The fields necessary to apply MuRAT can be checked in the workspace and the command window or displayed directly with the *verboseOutput* on. Figure 6.1 shows a run of Murat_test for a file stored in the sac_MSH folder, filtered at 6 Hz, smoothing eight samples per cycle.

6.3.2 $Murat\_testAll.m$ - creating a file to check all headers in the dataset

MuRAT checks that all the parameters in the headers for a dataset are populated. It produces error messages for the missing parameters, pointing at the corresponding waveform. Finally, it outputs the Excel file $DataHaeders.xsl$ in the results, providing an overview of all the required parameters. Fig. 6.2 shows this file after a run over the data folder sac_MSH. Missing parameters will be blank in the file. The origin and S-wave peaking (searched for the SAC variables o and $t0$) are missing. These are optional parameters if using P waves so that the code will work anyway.

The function can be used before running MuRAT. It provides a $DataHaeders.xsl$ file in the working folder that the user can check to set the variables in the header.



	Names	Origin	P	S	EvLat	EvLon	EvDepth	StLat	StLon	StElev
2	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610FL2-2000-01-08T14.57.03.997.SAC		21,27791595		46,1996994	-122,1903992	1,899999976	46,1962204	-122,3514862	1378
3	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610HSR-2000-01-08T14.57.03.997.SAC		19,4989624		46,1996994	-122,1903992	1,899999976	46,17427826	-122,1806488	1720
4	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610JLK-2000-01-08T14.57.03.997.SAC		20,0832901		46,1996994	-122,1903992	1,899999976	46,14706039	-122,1524277	1049
5	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610LVP-2000-01-08T14.57.03.997.SAC		22,92701912		46,1996994	-122,1903992	1,899999976	46,06594849	-122,4019318	1130
6	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610MTM-2000-01-08T14.57.03.997.SAC		22,45955849		46,1996994	-122,1903992	1,899999976	46,02532959	-122,2128677	1121
7	/Users/lucadesiena/Documents/MATLAB/MuRAT3D_update_2021/sac_MSH/20000108145722610SHW-2000-01-08T14.57.03.997.SAC		19,65478325		46,1996994	-122,1903992	1,899999976	46,19347	-122,236351	1425

Figure 6.2: Example file after running MuRAT for the MSH example. All the necessary parameters are checked. Notice the missing S-wave picking - the variable storing this parameter may be different from t0.

6.3.3 $Murat\_changeHdr.m$ - changing headers of single files

The previous analysis could discover some waveforms with missing or wrong header fields. The best procedure is to change these headers direct in SAC; however, users might not be familiar with the code, so $Murat\_changeHdr.m$ offers an alternative to change header fields:

```
f% function seism                    =    Murat_changeHdr(newfolder)
% CHANGES header of a file
%
% Input Parameters:
%       newFolder:               folder where you save the changed file
%
%   Output:
%       seism:                   new seismogram
%
```

The only input is the path to a folder where the user will store the SAC with a changed header. The user selects the SAC file to be changed through a GUI. Then, in the command window, the user is asked if the event, station or time fields are to be changed. Most often, the problem is with the time. The code assigns the origin to field $'o'$, the P-time to $'a'$, and the S-time to $'t0'$.

Chapter 7

Input files

MuRAT3 offers three input files as examples (Fig. 7.1). A new user needs to edit these files to run the code. The details of the files are given in each mlx file. The Mount St. Helen input file is shown in Appendix A. The new user can use this $Murat\_inputMSH.mlx$ file to learn what each parameter does; however, users soon switch to a more straightforward $.m$ file. The input files are divided into sections, whose inputs are described in the following. For a more detailed description, consult the $.mlx$ files or Appendix 1 in this Documentation.

7.1 The fields of the input

- GENERAL FIELDS:

  $Murat.input.dataDirectory$: Define the name of folders where data are located

  $Murat.input.label$: Defines the name of the folder where results will be stored

  $Murat.input.workers$: Number of cores if parallelised, leave empty for sequential code

- WAVEFORM DATA

  $Murat.input.originTime$: Defines the variable that stores the origin time or leave empty

  $Murat.input.PTime$: Defines the variable that stores the P-time

  $Murat.input.STime$: Defines the variable that stores the S-time or leave empty

  $Murat.input.PorS$: Defines if the analysis is for P- or S-waves

  $Murat.input.centralFrequency$: Defines the set of central frequencies analysed

  $Murat.input.components$: Defines the number of components

  $Murat.input.declustering$: Number used to divide the grid and declutter the dataset

- PEAK DELAY

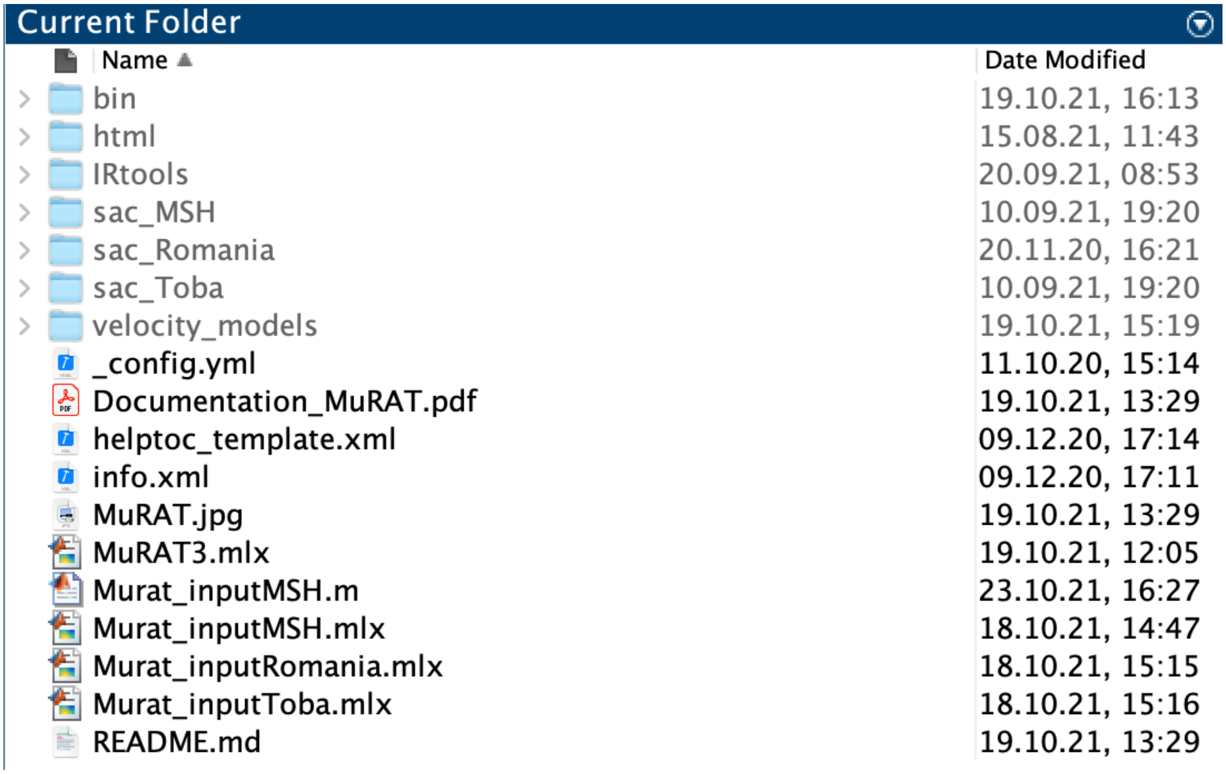


Figure 7.1: Input files are provided as $Murat\_input..mlx$ files. The standard release has three sample inputs (Mount St. Helens - MSH, Romania and Toba caldera).

$Murat.input.minimumPeakDelay$: Minimum peak delay allowed

$Murat.input.maximumPeakDelay$: Maximum peak delay allowed

- DIRECT WAVE ATTENUATION

  $Murat.input.spectralDecay$: Spectral decay of coda, for surface, diffusive or body waves

  $Murat.input.bodyWindow$: Window to measure body waves and noise in seconds

  $Murat.input.startNoise$: Seconds to start noise window from the start of recording

  $Murat.input.tresholdNoise$: Minimum signal-to-noise ratios allowed

- CODA ATTENUATION

  $Murat.input.lapseTimeMethod$: Defines the method used to measure coda waves

$Murat.input.startLapseTime$: Defines start of coda wave window

$Murat.input.codaWindow$: Defines the length of coda window

$Murat.input.maxtravel$: Set the max travel times after which you exclude traces

$Murat.input.albedo$: Measured Albedo from MLTWA

$Murat.input.extinctionLength$: Measured Extinction Length from MLTWA

$Murat.input.kernelTreshold$: Coefficient to make kernels more accurate or quicker to compute

$Murat.input.QcMeasurement$: Decide if using a 'Linearized' or 'Non-Linear' approach

$Murat.input.fitTresholdLinear$: Minimum fit accepted in the Linearized case

- GEOMETRY AND VELOCITY

  $Murat.input.origin$: Defines the origin of the spatial grid

  $Murat.input.end$: Defines the end of the spatial grid

  $Murat.input.gridLat$: Number of nodes across latitude

  $Murat.input.gridLon$: Number of nodes across longitude

  $Murat.input.gridZ$: Number of nodes across depth

  $Murat.input.sections$: Location of sections where you will get the maps

  $Murat.input.availableVelocity$: Decides if a 1D (0) or 3D (1) model is available

  $Murat.input.namev$: Defines the name of the velocity model

  $Murat.input.averageVelocityP$: Defines the average P-wave velocity

  $Murat.input.averageVelocityS$: Defines the average S-wave velocity

- INVERSION AND PLOTTING

  $Murat.input.inversionMethod$: Inverts with zero-orde Tikhonov or Iteratively

  $Murat.input.lCurve$: Decides if selecting the damping during computation

  $Murat.input.lCurveQc$: Damping parameters for different frequencies, Qc inversion

  $Murat.input.lCurveQ$: Damping parameters for different frequencies, Q inversion

  $Murat.input.sizeCheck$: Decides if doubling or making node space four times in checkerboard

$Murat.input.highCheck$: Decides the maximum inverse attenuation in the checkerboard

$Murat.input.lowCheck$: Decides the minimum inverse attenuation in the checkerboard

$Murat.input.spikeLocationOrigin$: Origin of the spike block

$Murat.input.spikeLocationEnd$: End of the spike block

$Murat.input.spikeValue$: Value of attenuation in the block

Each input is necessary, but the user will change only some of them repeatedly. Here, I give details on each group.

7.1.1 GENERAL FIELDS

These are set initially and never changed, except $Murat.input.workers$. The first run should always be sequential (set it to **[]**). The sequential run must be used to debug the code. While MuRAT has several embedded error messages, stopping the run is often necessary, especially during data processing (see $Murat\_data$). The debug function in Matlab does not allow pinpointing the incorrect waveform with parallelised processing. However, it is recommended to perform the final runs in parallelised mode: the parallelised processing reduces computational time from ~ 15 s to ~ 4 s for the MSH dataset.

7.1.2 WAVEFORM DATA

You can check the SAC header requirements in chapter 6. Once set, the header variable must be the same for all files (e.g., variable *'a'* for P-wave picking or *'o'* for the origin). The number of components is defined at the start. The code allows a single analysis for multiple frequencies with bandwidth $\frac{1}{3}$ of the central frequency. Finally, the declustering variable divides the grid into smaller cells to select the best earthquake waveforms in the cell, avoiding different repeated measurements that increase residuals. It is recommended to test the code first without declustering.

7.1.3 PEAK DELAY

See chapter 3 for additional details. Setting the minimum and, especially, the maximum allows for avoiding tracking phases unrelated to the direct wave arrival. Sometimes tracking these phases, typically surface waves generated by a strong impedance contrast at the surface, is helpful to map changes in topography and geomorphology (De Siena et al., 2016; Gabrielli et al., 2020).

7.1.4 DIRECT WAVE ATTENUATION

See chapter 2 for additional details. The spectral day has minimal effect on variations in space, but it can affect absolute values of total attenuation. $Murat.input.bodyWindow$ must be small enough to assume still a ray approximation but large enough to mitigate changes in radiation pattern, the primary trade-off of the coda-normalisation method.

$Murat.input.startNoise$ depends on how much noise you have at the start of your recordings. The code will measure noise from this point on the window set for the body wave and use it to compute signal-to-noise ratios. All waveforms below $Murat.input.tresholdNoise$ are discarded.

7.1.5 CODA ATTENUATION

See chapter 4 for additional details. There are three methods to measure coda: using a constant start time (**'Constant'**), the peak amplitude (**'Peak'**) and a number multiplied by the travel time (**'Travel'**). The standard practice uses the last option (e.g., $2t_S$). However, coherent waves are often still in the corresponding window (see chapter 9) (Calvet et al., 2013). The **'Peak'** is used when employing active data in volcanic, as the waveform becomes rapidly diffusive (Wegler, 2003).

If the user did not measure albedo or extinction length, they could use published values for the area or the crust. $Murat.input.kernelTreshold$ must be set to find a compromise between the accuracy of the kernels and computational costs. Setting it to 1 solves the forward kernels on the same grid used in the inversion: this is typically too coarse, e.g., to interpolate the poles of the kernels at the source and station. The minimum value is set to two, with 4/8 providing the most accurate results.

While the standard practice is to linearise the inversion by taking the logarithm of the envelope, the user can choose a non-linear approach (Napolitano et al., 2020). In the first case, the minimum accuracy is set with $Murat.input.fitTresholdLinear$, representing the Pearson coefficient relative to a line. The advice is to set it to a very low value (e.g., 0.1) as this measurement will be used to weigh the inversion. In the non-linear case, the weight is given by the PDF of the misfit.

7.1.6 GEOMETRY AND VELOCITY

The inversion grid's extension and number of nodes are often changed throughout the analysis because of the need for testing and avoiding spurious earthquakes. MuRAT inverts on a regular grid defined in latitude, longitude and depth. If using a 1D model ($Murat.input.availableVelocity$=0), the user can input depths in km; otherwise, using a 3D model ($Murat.input.availableVelocity$=1), depths are in meters. The text file of the 3D velocity model must replicate the structure of $modvMSH.txt$ for the 1D that of $iasp91.txt$. Both files are inside the **velocity_models** folder. The user can set average velocities for P and S waves for standard calculations.

7.1.7 INVERSION

The regtools ($Murat.input.inversionMethod$ = **'Tikhonov'**) and IR Tools (= **'Iterative'** or =**Hybrid'**) are the two packages included in the code and necessary to invert for direct and coda attenuation. After the first run of MuRAT, a new folder (**Test**) will provide several figures that show either the residual vs norm function or a cost function that must be minimised (Aster et al., 2005). The user dampens the inversion by selecting the corresponding parameter for each frequency.

The other inputs define the multiplication factor of the checkerboard test relative to node spacing (2 or 4) and the values of the alternating checkerboard anomalies. The inputs for the spike test define its location and value.

Chapter 8

Output files

MuRAT3 has several output folders containing files that give you inside into the quality of the analysis (Fig. 8.1). Also, after running, the code creates fields related to the output in the $Murat.mat$ file.

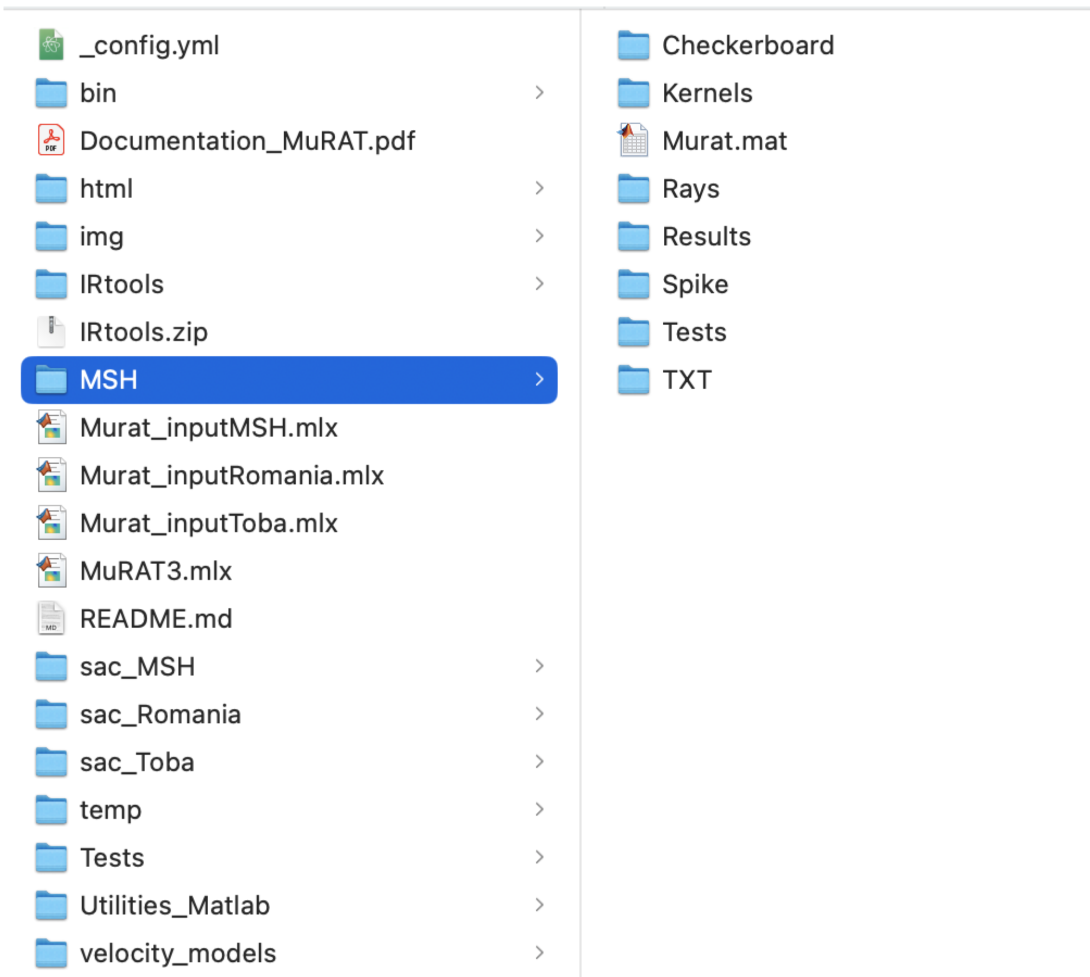


Figure 8.1: Example structure of the results after a run on the MSH dataset.

8.1 The folders and files after running the code

- CHECKERBOARD:

  $Qfolder$: Contains the frequency-dependent .fig and .tif files showing the checkerboard test inputs and outputs for the Q inversion across the sections defined in the input.

  $Qcfolder$: Contains the frequency-dependent .fig and .tif files showing the checkerboard test inputs and outputs for the Qc inversion across the sections defined in the input.

- RAYS AND KERNELS

  $Kernels$: The folder contains all the frequency-dependent kernel figures for all methods.

  $Rays$: The folder contains all the frequency-dependent ray figures for all methods.

- RESULTS

  $Parameter$: Contains the frequency-dependent .fig and .tif files showing the results of the parameter space analyses across the sections defined in the input.

  $PeakDelay$: Contains the frequency-dependent .fig and .tif files showing the results of the peak delay analysis across the sections defined in the input. The same figures are provided, removing cells with less than 5% of the total rays.

  Q: Contains the frequency-dependent .fig and .tif files showing the results of the Q analysis across the sections defined in the input.

  Qc: Contains the frequency-dependent .fig and .tif files showing the results of the Qc analysis across the sections defined in the input.

- SPIKE

  $Qfolder$: Contains the frequency-dependent .fig and .tif files showing the spike test inputs and outputs for the Q inversion across the sections defined in the input.

  $Qcfolder$: Contains the frequency-dependent .fig and .tif files showing the spike test inputs and outputs for the Qc inversion across the sections defined in the input.

- TESTS

  $Clustering$: A figure showing the original rays (black) and those selected after declustering (red).

$LCurve$: A folder that contains the frequency-dependent .fig and .tif files showing the L curves and cost functions for the chosen inversion.

$PeakDelay$: Test for the reliability of the peak delay method when applied to data.

Q: Test for the reliability of the Q method when applied to data and Picard condition for the inversion.

Qc: Test for the reliability of the Qc method when applied to data and Picard condition for the inversion.

$Qc_vs_frequency$: Relation between Qc and frequency over the span chosen.

- TXT

  $DataHeaders.xls$: Excel file containing header infos from SAC files

  $parameters$: Frequency-dependent text files allocating parameter number to space.

  $peakdelay$: Frequency-dependent text files to show peak delay results. The fourth column is the peak delay, while the fifth is ray crossing.

  Q: Frequency-dependent text files to show Q results. The fourth column is the inverse Q. The fifth is all zeros. The sixth and seventh (eighth and ninth) columns are the inputs and outputs of the checkerboard (spike) tests.

  Qc: Frequency-dependent text files to show Qc results. The fifth is all zeros. The sixth and seventh (eighth and ninth) columns are the inputs and outputs of the checkerboard (spike) tests.

8.2 The Murat .mat file

The Murat.mat file saves all information in the $input$ and $data$ fields. The $input$ field recaps all the input the user has set for the run. The $data$ field contains all fields obtained during the run. Among these, the users are likely to check $Murat.problemPD, .problemQ, .problemQC$.These structures include the paths to files that did not satisfy the requirements for the analyses, where each column corresponds to a figure. The user can check or remove these files from the study.

Chapter 9

Tests

The code provides test figures for the three methods. These are all in the **Tests** folder, created by the code in the **Results** folder. These tests are divided into three categories:

- **Analyses** of method data and inversion quality;
- **Curves** that allow defining the best damping parameters for the inversion;
- **Plots** of the behaviour of coda attenuation vs frequency and velocity models.

9.1 Analyses

The *Analyses* figures (files $*.analysis\_f\_Hz$) are the most crucial output in MuRAT. They allow for assessing the data quality and inversion (except for the peak delay, which is only regionalised).

9.1.1 Peak delays

As described in Chapter 3, the hypothesis behind peak-delay imaging is that peaks increase vs travel times. Figure 9.2 shows the $PD\_analysis\_frequency$ figures for the three sample datasets.

These figures allow you to compare the differences (and problems) from peak-delay analysis in different frequency bands. The Mount St Helens example (Figure 9.1) shows how the 3 Hz frequency is affected by surface waves acting at short travel times. This effect is due to unconsolidated materials from the 1980 debris avalanche (Gabrielli et al., 2020). Analyses are only possible above 6 Hz.

The user needs to check the fit of the peak-delay behaviour to a line to apply the regionalisation procedure, as explained in Chapter 3.

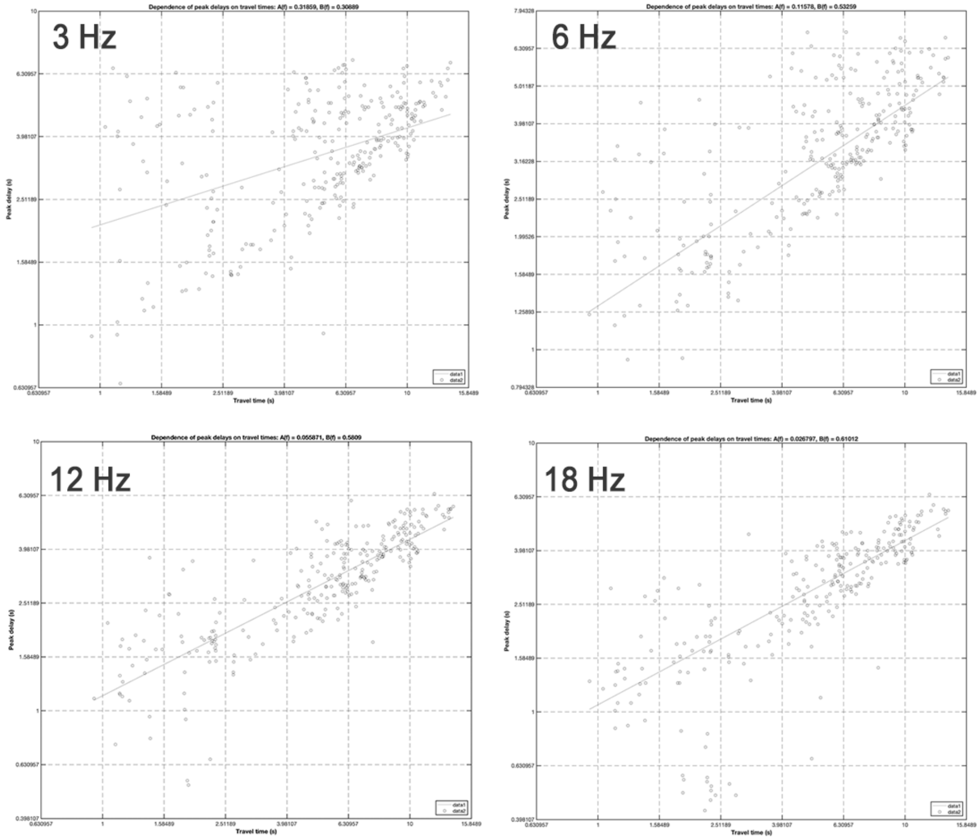


Figure 9.1: Comparison between peak delays for the four frequencies and the Mount St Helens dataset.

9.1.2 Coda attenuation

As explained in Chapter 4, **coda attenuation must be constant with increasing ray length (Calvet et al., 2013)** to interpret values of Q_c as absorption. MuRAT outputs this simple variation for a first-order check on the data. A constant Q_c does not mean the field is diffusive but indicates that equipartition is likely to occur. One can use the results of an *MLTWA* analysis (Fehler et al., 1992; Del Pezzo et al., 2001; Del Pezzo, 2008) for better calculations. Codes like Qopen (Eulenfeld and Wegler, 2016) precisely measure mean free path and transport mean free path, providing better constraints on the onset of diffusion. However, their results are generally equivalent to those of MLTWA (van Laaten et al., 2021).

We perform the inversion using the external packages regtools (Hansen, 2007) and their evolution for iterative regularisation, the IRTools (Gazzola et al., 2019). The Picard condition gives a first glance at how many coda-attenuation parameters will be solved by the inversion in space (see Aster et al. (2005) for a review).

9.1.3 Direct-wave attenuation

The coda-normalisation method (chapter 2) **requires the use of direct waves that decay linearly with travel time (hypocentral distance)**. The coda-normalisation analysis (Fig. 9.4) shows how its normalised logarithm decays for increasing travel time (upper panel). The data (black circles) are compared to the analytical forward with average attenuation measured on the data (red stars). The middle panel shows the behaviour of direct energy Without normalisation and correction for the velocity model. The inversion is tested with the usual Picard condition, showing that, for the chosen parametrisation, more than half of the parameters are solved.

9.2 Damping

The inversion for Q_c and Q requires setting damping and smoothing parameters for each frequency. The L_curve* figures show the relation between residual and norm of the solution if using the *regtools* (Hansen, 2007). Nevertheless, it is more sensible to employ the *IRTools* (Gazzola et al., 2019), as they offer much better stability for large inversion problems and allow visualising a cost function considering both damping and smoothing (Rawlinson and Spakman, 2016). In Fig. 9.5, these two plots and corresponding cost functions are shown for the MSH dataset at 6 Hz. The Q_c inversion is way more unstable, as hinted by the Picard condition (Fig. 9.3), as shown by the difference between minimisation parameters (two orders of magnitude). The red rectangle indicates that damping and smoothing are the same for the Q inversion, which solves most model parameters (Fig. 9.4, Picard condition). However, in both cases, a minimum is reached.

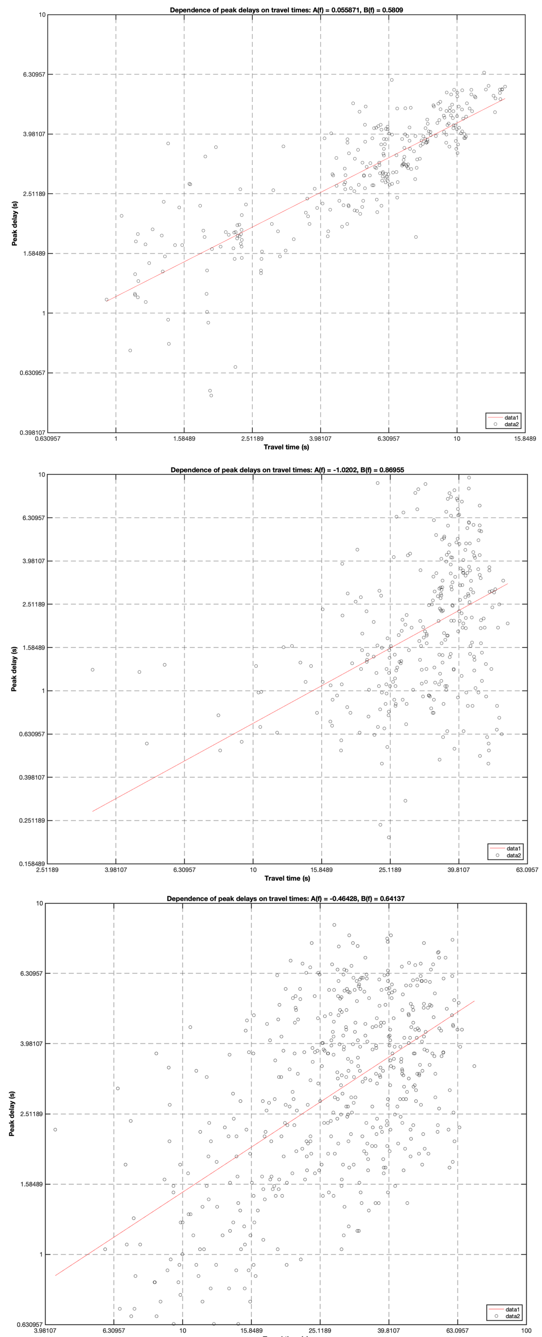


Figure 9.2: Comparison between peak delays for the three sample datasets.

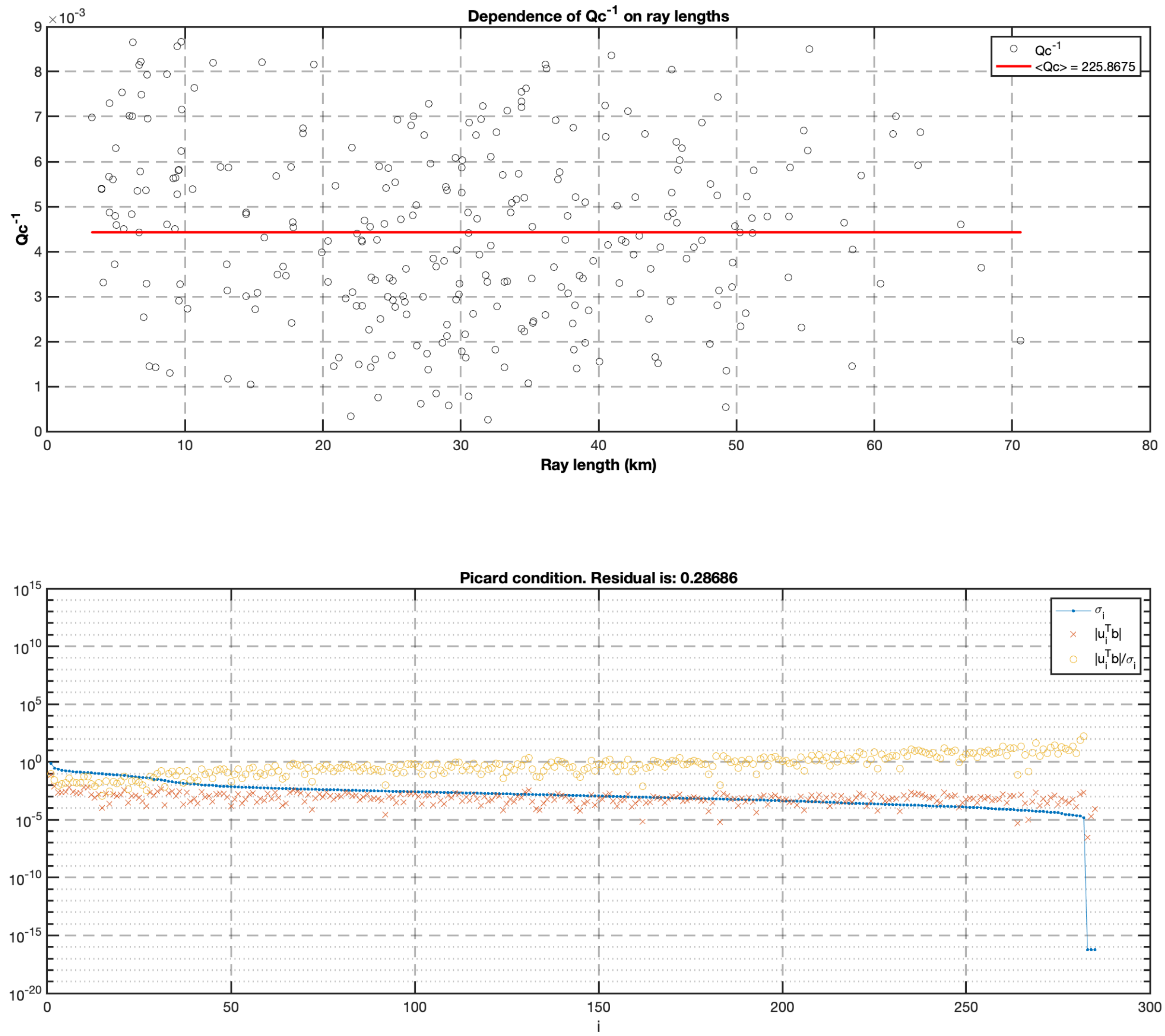


Figure 9.3: Coda analysis plot for the MSH dataset at 6 Hz. The upper panel shows that Q_c is constant with the ray length computed through ray tracing. The lower panel depicts the result of the Picard condition (Aster et al., 2005): only half of the parameter models are solved under the weakest hypothesis (blue and red lines cross around the 150th parameter). Only one-tenth of the parameter models are solved if we also consider the uncertainty of data. σ are the singular values.

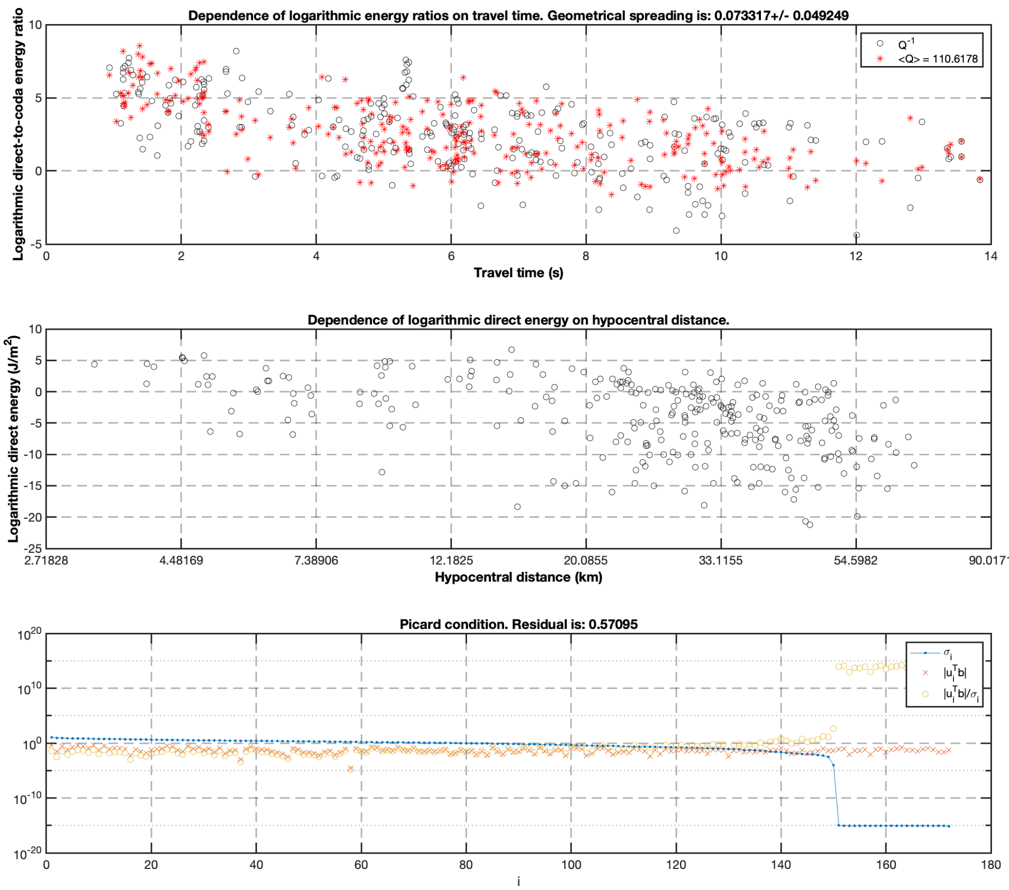


Figure 9.4: Analysis of the data and inversion results for the coda-normalisation inversion at MSH for the 6 Hz frequency band. The upper panel shows the decay of the logarithmic energy ratio with travel time for the data (black circle) and the theoretical behaviour with average attenuation. The middle panel shows the dependence of logarithmic energy on hypocentral distance. The Picard condition (bottom panel) gives a first glance at how many coda-attenuation parameters will be solved by the inversion in space (see Aster et al. (2005) for a review).

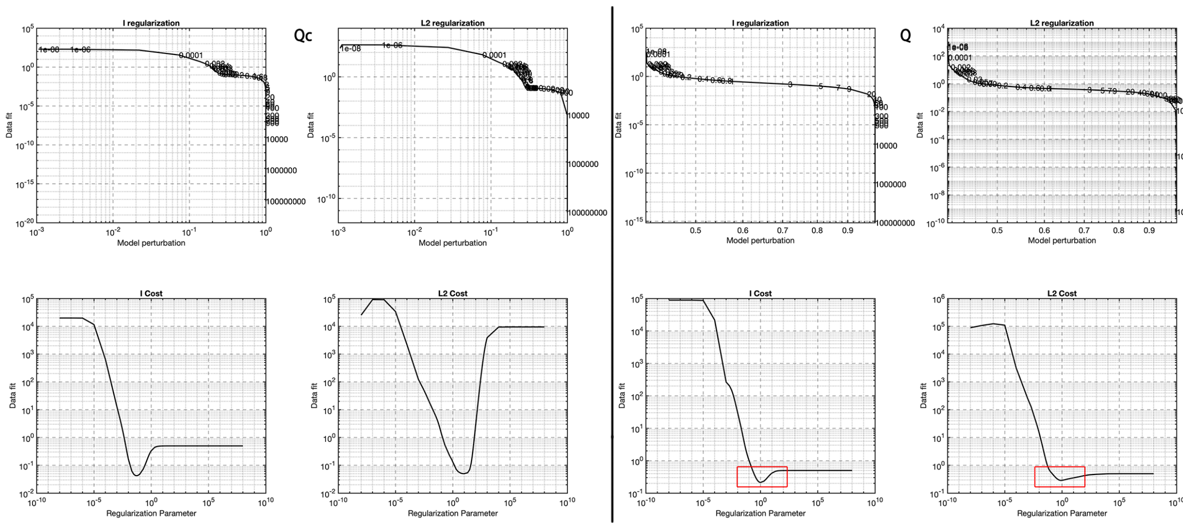


Figure 9.5: Output of the IRTools showing the relation between data fit and either the solution or its smoothness (upper panels, equivalent to the L_curves) for the MSH dataset at 6 Hz. The lowermost panels show the corresponding cost function to be minimised. In the case of Q, the cost minimisation parameter (red rectangle) is the same for both conditions.

Chapter 10

Rays and Kernels

While scattering and total attenuation variations are based on a ray-bending technique (Block, 1991), MuRAT includes 3D sensitivity kernels (Del Pezzo et al., 2018) for modelling seismic absorption in space.

10.1 Ray tracing

MuRAT ports the original Fortran code for ray bending from Block (1991) into Matlab. The code is a classical ray bending, starting from the source and receiver locations and bending the ray to minimise travel time inside the input velocity model. The user can look at the function $Murat\_tracing$, which *traces* the ray in the velocity model. At the start of the function, two optional parameters set the maximum number of bending iterations and points allowed.

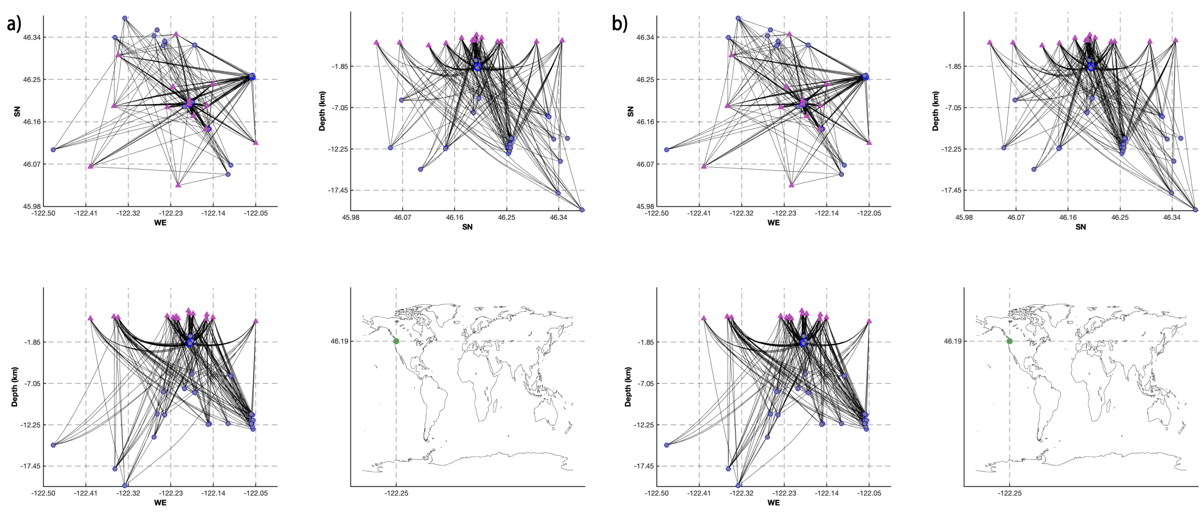


Figure 10.1: Rays of the coda-normalisation (a) and peak delay (b) methods at 3 Hz for the MSH dataset.

Their values (*maxit = 100* and *maxpoints = 10000*, respectively) have been set after several trials and generally work just fine at the crustal scale. The rest of the procedure includes the original function that averages velocities on the regular grid and calculates velocity gradients, informing the bending process (Block, 1991). The functions have been tested against standard Matlab functions and are generally three to ten times faster.

MuRAT pairs to the ray-bending approach some utilities optimised for speed that measure ray lengths across the inversion grid. These lengths are used both in the standard coda-normalisation method and as a weight in the peak-delay regionalisation (Chapters 2-3). The user can look at the function $Murat\_rays$, which calls both procedures, to see that the procedure needs the velocity model, a propagation grid, and source/station locations as input. MuRAT calls $Murat\_segments$ within the function, calculating segment lengths that it includes directly in the coda-normalisation matrix (see chapter 2) and peak-delay matrix. The corresponding rays are finally shown in the folder **RaysKernals**. As the data selection is different for coda-normalisation and peak-delay, the code outputs two different sets of **Rays_*** figures (Fig. 10.1). The ray tracing works for both the small volcanic scale (like at MSH, Fig. 10.2a), mantle scale (Fig. 10.2b) and continental-oceanic settings (Fig. 10.2c).

10.2 Kernel implementation

MuRAT implements the computational kernels described by Del Pezzo et al. (2018) in the tomographic procedure described by De Siena et al. (2017a). The kernels (Fig. 10.3) are calculated with a call to the function $Murat\_kernels$, using the albedo and extinction lengths input in the code as input. Panel a) shows the entire computational kernel computed up to the laspe time equal to the starting lapse time plus half of the coda window.

MuRAT produces no MLTWA analysis, yet the user needs to set albedo and extinction length. The values in the input files are average for the crust; better measurements are likely available for the area under study.

10.3 Geographic and cartesian coordinates

MuRAT has all inputs in latitude and longitude but needs conversion to cartesian. This conversion is performed mainly in $Murat\_checks$ and through other functions within the bin. This approach means we rely on the functions $distance$ and $reckon$ to create, e.g., velocity models for ray propagation or switch between cartesian and geographic outputs.

The implementation in cartesian coordinates is relatively straightforward and already implemented by works at the 10s of meters and laboratory scales - see ***Di Martino et al. 2022*** and ***King et al. 2022***.

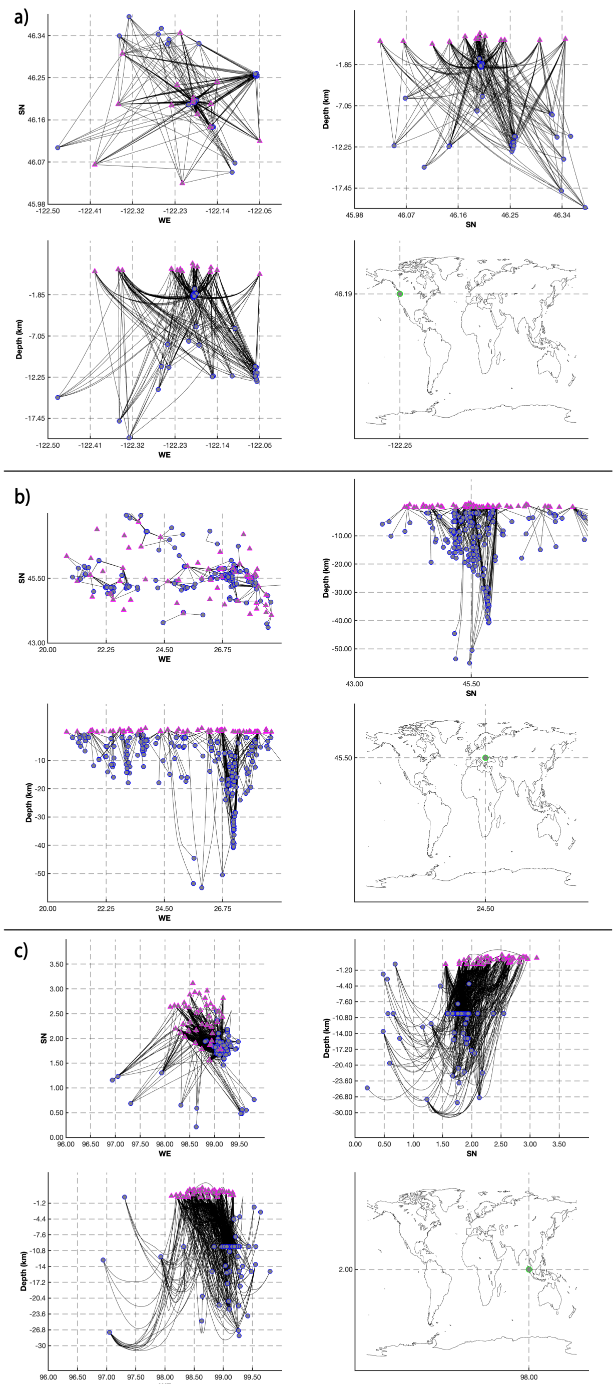


Figure 10.2: Rays of the coda-normalisation method at 3 Hz for the three sample datasets: a) MSH; b) Romania; c) Toba.

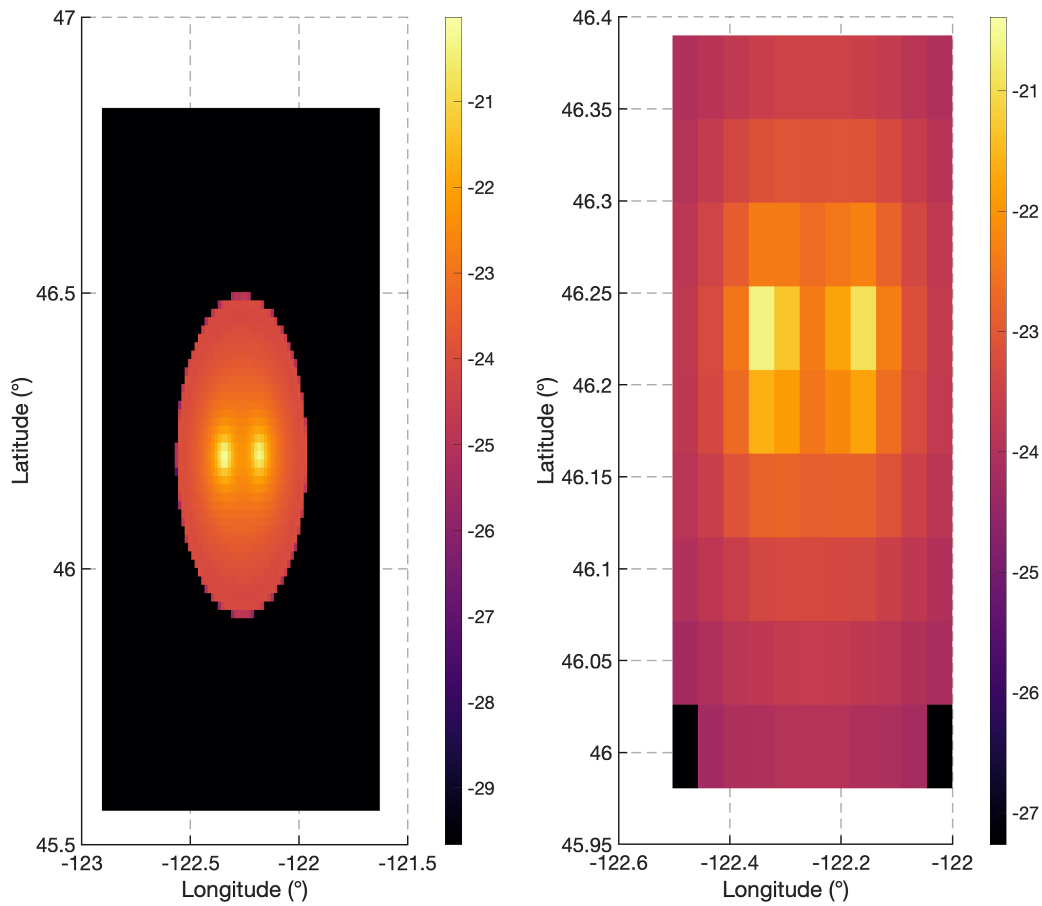


Figure 10.3: Kernels for coda-attenuation imaging at 3 Hz for the MSH dataset. The full kernels are shown as a section in the 3D space (a) and in the grid chosen for the inversion (b). Notice that sensitivity is always highest at the source and receiver. The colour scale is logarithmic. The lapse time is 22.5 s.

Appendices

Appendix A

MuRAT input file - Mount St Helens

Table of Contents

INPUT MuRAT3D - MSH

This is an input file for the program Multi-Resolution Attenuation Tomography (MuRAT), version 3. It refers to the following area:

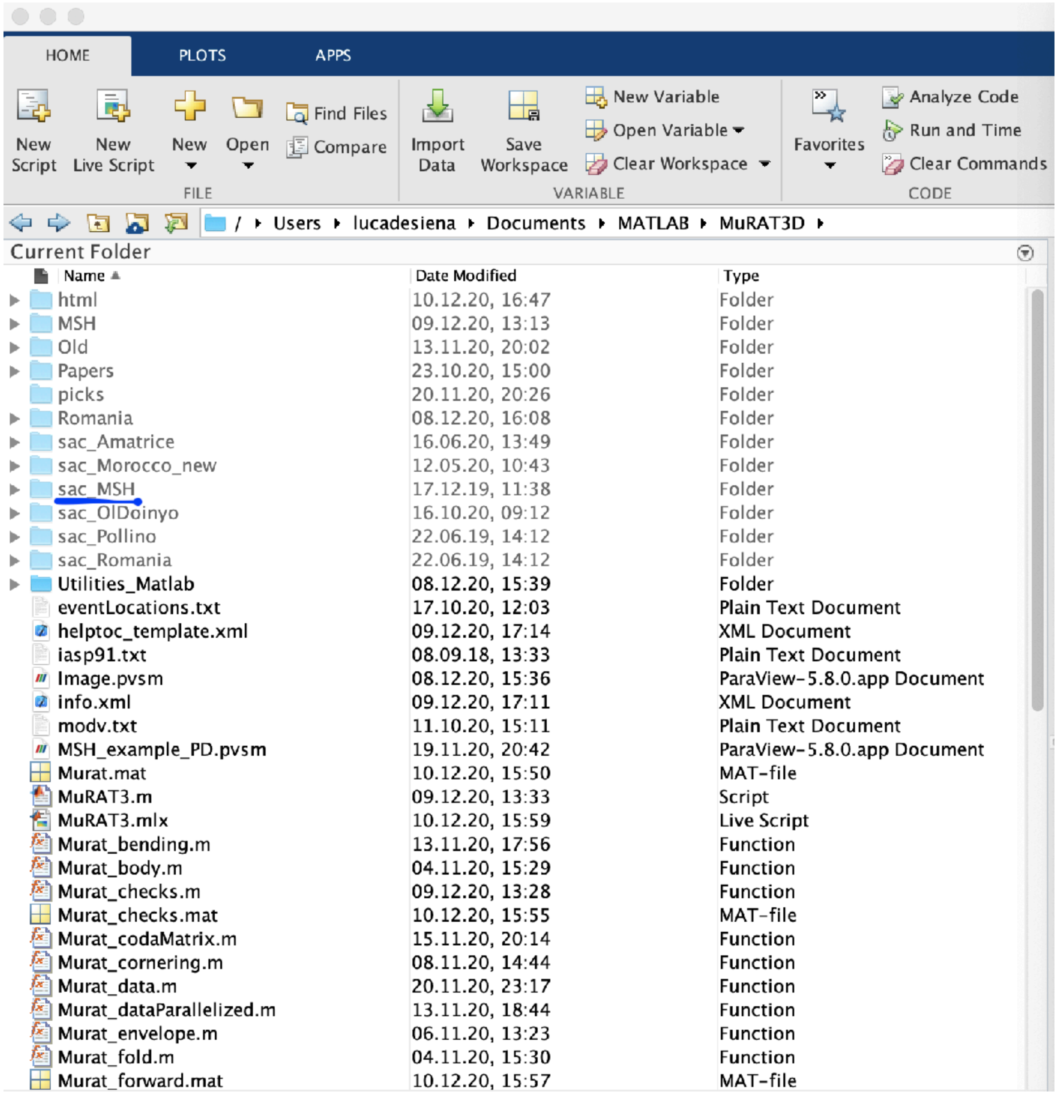
```
MOUNT ST HELENS VOLCANO
```

The working directory is the folder where you downloaded MuRAT: move it anywhere in your system.

GENERAL FIELDS

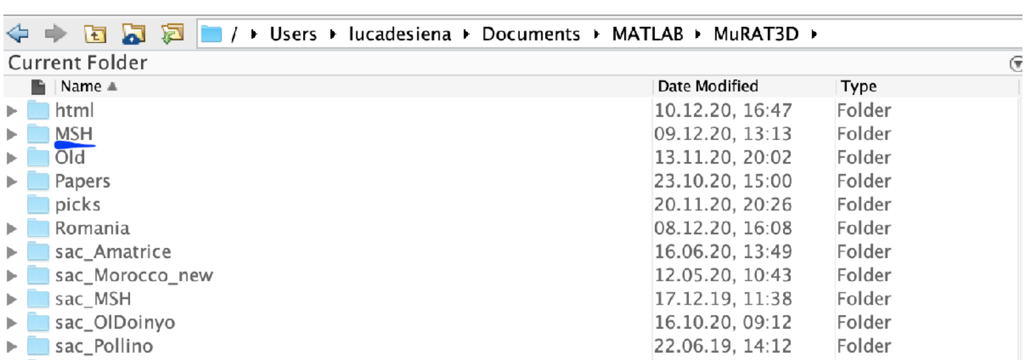
The user needs a data directory containing *.sac* files (beware of the difference between *.sac* and *.SAC*). If you want to use two or three-component recordings, you have to store them into a single folder, but the three components must be in the exact order **E,N,Z**. For an example see Toba. For this example we will use data in the **sac_MSH** folder. Inside this folder, only vertical (*Z*) seismograms are stored:

```
Murat.input.dataDirectory            =    'sac_MSH';
```



Best practice is to create a new folder to store your results. The code will create this folder inside the working directory. Specify the name of the folder that will store text files and figures, and will appear in your working directory:

```
Murat.input.label                    =   'MSH';
```



In MuRAT3D you can choose between a sequential or parallelized forward loop. In the parallelized case, just set a number of workers (cores). In this example, we are working with the parallelized code and a computer with 8 cores. Otherwise, set *Murat.input.workers* as empty (**[]**).

```
Murat.input.workers                    =    [];
```

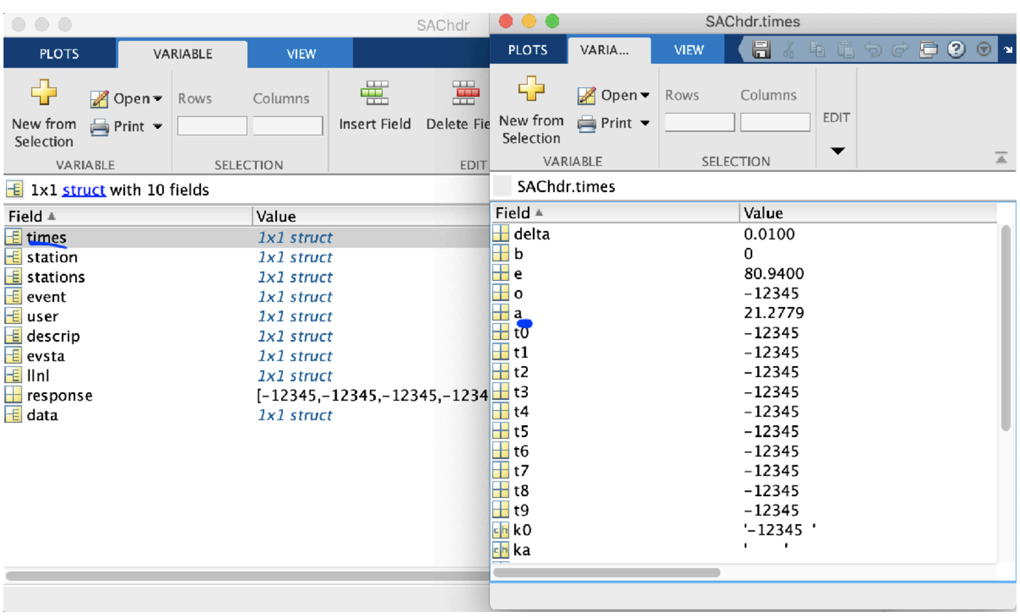
WAVEFORM DATA

Set all the variables required by data processing. This includes data choises, as setting the name of the variables in SAC and all the attributes that are needed for the three kinds of imaging. The routines for loading the files have been mostly created by Zhigang Peng and co-workers and downloaded from their Introduction to SAC webpage.

MuRAT3D is designed to work with *sac* files only, so it is necessary to set the name of the variable containing their peakings and zero time. Files have to have populated headers. Set the name of the variable containing the origin time, P-wave peaking and S-wave pickings - in this example there is only the mandatory P-wave picking. If you don't have the origin time and S-wave times at hand you can mark them as empty (**[]**):

```
Murat.input.originTime                 =    [];
Murat.input.PTime                      =    'a';
Murat.input.STime                      = [];
```

SAChdr is a structure that contains all the fields you saved in the SAC haeder. In this example the P-wave picking has been stored inside variable *a.* You always find these variables in the *times* subfield:



Then choose the coherent phase you are analyzing - P-(**2**) or S-(**3**). In our case it is P- as we have no S-wave picking:

```
Murat.input.POrS                      = 2;
```

You need to set the central frequencies (Hz) according to your spectrograms. General practice is to vary it across your spectra (see De Siena et al. 2016, EPSL) for absorption and scattering mapping or focus on a given frequency (De Siena et al. 2014, JGR) for direct-wave attenuation imaging. Here, they cover the interval [1.5-24] Hz:

```
Murat.input.centralFrequency          = [3 6 12 18];
```

You can work with 1 vertical or horizontal (*1*), 2 horizontal (*2*) or three components(*3*). If using more than one component, the order *MUST BE: WE, SN, Vertical or SN, WE,* Vertical. In this example we work with the vertical component only:

```
Murat.input.components                = 1;
```

Finally, you can opt to decluster your data events. The code will divide the inversion grid by the following factor and select the best earthquake located in the block among all others. Set it to empty if you want to opt out (**[]**).

```
Murat.input.declustering              = 5;
```

PEAK DELAY

Scattering (peak delay) measurements rely on the existence of coherent waves. Peak delay is a standard measurements of forward scattering in regional scale-mapping since Takahashi et al. 2007, GJI. Here, we approach the problem by adding a ray-tracing strategy to the system, assuming that the sensitivity follows the seismic ray. Also, we bring it to 3D and apply it to P-waves (see De Siena et al. 2016, EPSL).

To do so we need to set a minimum peak delay (*s*) considering scattering in the area and frequency. If using P wave this is important, as they can be sometimes more energetic than S-waves, biasing our measurements. We also need to set the maximum peack-delay (*s*) to avoid picking surface waves.

```
Murat.input.minimumPeakDelay          = 0.1;
Murat.input.maximumPeakDelay          = 7;
```

DIRECT WAVE ATTENUATION

Total attenuation (inverse Q) measurements also relies on the existence of coherent waves. Total attenuation with the coda-normalization method is today astandard in volcano tomography (Del Pezzo et al. 2006, PEPI, De Siena et al. 2010, JGR; De Siena et al. 2014, JGR, Prudencio et al. 2015, Prudencio & Manga, 2020.

The method relies on the measurements of both direct and coda wave energy. Therefore we need to set the spectral energy decay of the coda wavefield. The spectral decay has to be set for prdominance of 2D surface (0.5), diffusive (1.5) or body waves (2). Here it is it is the first case:

```
Murat.input.spectralDecay             = 0.5;
```

Then we set the length of the window used to measure direct-wave energy, trying to smooth radiation pattern effects. The code uses the lame length to measure noise energy. Discussions on the topic can be found in De Siena et al. 2009, PEPI and De Siena et al. 2010, JGR, but the choise is as usual dependent on data:

```
Murat.input.bodyWindow                   = 1;
```

We also need the start of the window used to measure noise a few seconds after the start of the recording (in seconds).

```
Murat.input.startNoise                   = 5;
```

The coda-to-noise energy ratio is used by the weighted inversion. Here you set the minimum accepted energy ratio.

```
Murat.input.tresholdNoise                = 3;
```

CODA ATTENUATION

Coda attenuation (inverse Qc) is measured from the decay of the coda with lapse time from the origin time of the earthquake, and requires energetic scattering. Qc is a well know parameter for assesing tectonic structures (Sato et al. 2012, Springer). In recent years it has been used as an imaging attribute at the regional (Calvet et al. 2013, Tectonophysics; Borleanu et al. 2017, Tectonophysics), fault (Napolitano et al. 2020; Sketsiou et al. 2020, PEPI) and, especially, volcanic scales (Prudencio et al. 2013a, GJI; Prudencio et al. 2013b GJI; De Siena et al. 2016, EPSL; De Siena et al. 2017, GRL; Gabrielli et al. 2020, GJI).

Lapse times and sensitivity kernels

In MuRAT3D, we use the full 3D computational kernels devised by Del Pezzo et al. 2018, Geosciences in the inversion approach proposed first by De Siena et al. 2017, GRL.

Here you choose the start of the window used to measure coda wave energy and model kernels. The starting time of the coda window can be set directly in seconds ('**Constant**'), from the envelope peak ('**Peak**'), or depending on the P- or S-wave travel time, for example twice the S-wave travel time ('**Travel**'). If the chosen method is **Constant**, set start and length of the window.

```
Murat.input.lapseTimeMethod              =    'Constant';
```

If the chosen method is **Constant**, set the start of the window in seconds after the origin time. It it is **Peak**, set it to **[]**. If it is **Travel** select *Murat.input.startLapseTime* as the moltiplicative factor of the pahase you are using (e.g., **2** or **3**). Avoid this method if you do not know the S-wave time.

```
Murat.input.startLapseTime               =    15;
```

Finally set the length of the coda window in seconds. The true lapse time at which we calculate the kernels is half of the window. The window is also used (after normalizing for its length) in the coda normalization method.

```
Murat.input.codaWindow                   =    15;
```

Set maximum travel time if you want to exclude traces beyond a certain value, else put [Inf].

```
Murat.input.maxtravel                   =   13;
```

The **MLTWA** is the standard method to find the average parameters necessary to calculate the kernels. It provides albedo and extinction length:

```
Murat.input.albedo                      =   0.5;
Murat.input.extinctionLength            =   0.02;
```

As the kernels are computational intensive and require a full matrix of nodes to avoid singularities, we also use a computational factor to reduce the computational time. This number divides the input grid, meaning that higher numbers give more precise results at the expene of computational time. Minimum is **1**. A figure will output the kernel in this grid.

```
Murat.input.kernelTreshold              = 2;
```

Measurement of Qc

MuRAT3D implements either a linearised approach or a grid search approach to measure Qc. The linearised approach is the standard proposed first by Aki (e.g., Havskov et al. 2016, BSSA) to best fit Qc after taking the logarithm of the energy. The uncertainties are derived from the simple minimum R-squared (fitTresholdLinear) and needs to be defined by a number between 0 and 1. It is advisable to set a minimum of 0.1.

The non linear approach models energy data measured on one-second windows across the envelope and minimizes the difference between data and model with a 1D grid search algorithm (Napolitano et al. 2020). Uncertainties are given by the experimental probability density function of the misfit. In both cases, uncertainties play as a weight in the final inversion. In the second case, leave the fitTresholdLinear = **[]**.

The user needs to choose between the two options **'Linearized'** and **'NonLinear':**

```
Murat.input.QcMeasurement               =   'Linearized';
Murat.input.fitTresholdLinear           = 0.1;
```

GEOMETRY AND VELOCITY

This section sets the details of the inversion grid and availability of velocity model. In MuRAT3D the coordinates of the model are in lat/lon, then they get converted in km. The vertical is in altitude above sea level. The velocity model can be 1D or 3D - if 3D all points must be given in lat/long formats. You start by setting the origin and end points of your inversion grid.

```
Murat.input.origin                      =   [45.9805 -122.5030 3350];
Murat.input.end                         =   [46.3900 -122.0000 -22650];
```

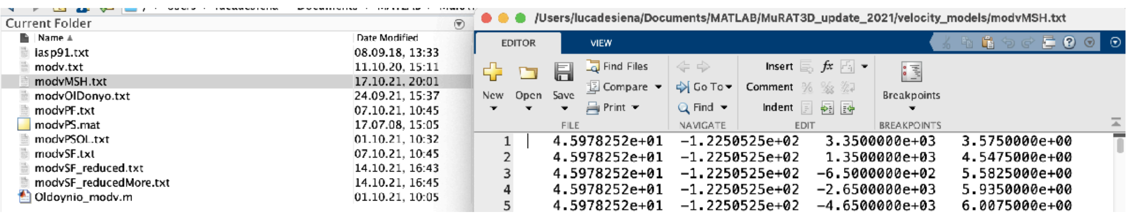
Then you need to set the number of nodes in the three directions. This is obviously dependent on the scale of your area. You will be playing a lot with this to test your effective resolution capabiities.

```
Murat.input.gridLat                     =   10;
Murat.input.gridLong                    =   12;
Murat.input.gridZ                       =   6;
```

MuRAT3D now saves everything in Paraview format (.VTK) but some checks in Matlab can be useful. All these sections are saved in the *Label* folder. You will see all of your figures on three sections cutting the models WE (degrees), SN (degrees), and horizontally (meters or km) at:

```
Murat.input.sections                    = [46.12 -122.20 -1000];
```

With this version of the code you are always using an underlying velocity model: the 3D is either unavailable(***0***) or a vailable (***1***) velocity model. For the 1D case MuRAT provides you *iasp91.txt*, the standard IASPEI velocity model and expands it to a false 3D. However, a standard crustal model is generally available everywhere on the Earth, so use that - but change it to the same format as the file provided, first column is depth, second is distance from the centre of the Earth, then third and fourth are P- and S-wave velocity. Store the file in the folder **velocity_models**. See the Romania example file. At Mount St. Helens, we use the 3D local earthquake tomography model of Waite and Moran, 2009,JVGR, (*modvMSH.txt*), stored in the folder **velocity_model:**



whose coordinates are Lat/Long/Altitude/Velocity in meters. So we set:

```
Murat.input.availableVelocity           = 1;
Murat.input.namev                       =    'modvMSH.txt';
```

Even if we set the velocity model we still need the average crustal velocities if you have no info of origin time. It is highly recommended you have the origin in the haeder, at variables **'o'**!

```
Murat.input.averageVelocityP            =    6;
Murat.input.averageVelocityS            =    3;
```

INVERSION

The code implements either a standard Tikhonov inversion based on singular value decomposition (**'Tikhonov'**) and an iterative conjugate graduate least square inversion minimizing both model norm and Laplacian (**'Iterative'**). For the first, we rely on the regtools Matlab suite from Per Christian Hansen. The second uses the IR TOOLS for iterative regularization. Copies of both packages are linked to MURAT3:

```
Murat.input.inversionMethod             =    'Iterative';
```

As you need to select the damping you can choose to output the L curves between residual and norm length (Aster et al. 2013 - case **'Tikhonov'**) or the cost functions (case **'Iterative'**). Set either ***1*** or ***0***. In the latter case, set the damping for both Qc and Q inversions. You will likely test many different damping parameters and likely want to avoid seeing the same figure again, so you can just declare the damping for Qc and Q after looking at the figures. Remember doing it for each frequency.

```
Murat.input.lCurve                      =    0;
Murat.input.lCurveQc                    =    [0.1 0.1 0.05 0.02];
```

```
Murat.input.lCurveQ                    =   [1 1 0.5 0.2];
```

A great reference for the best sort of testing is Rawlinson & Spakman, 2016. If you want to test you results you need to create a checkerboard. The size of the checks can be twice (*2*) or four times (*4*) node spacing:

```
Murat.input.sizeCheck                  =   2;
```

with each cell having alternating values of attenuation:

```
Murat.input.highCheck                  =   0.02;
Murat.input.lowCheck                   =   0.001;
```

In MuRAT3D you can also set the origin and end locations of a spike as well as its attenuation value:

```
Murat.input.spikeLocationOrigin        =   [46.05 -122.40   0];
Murat.input.spikeLocationEnd           =   [46.12 -122.30 -6000];
Murat.input.spikeValue                 =   0.02;
```

Bibliography

W. G. Akande, L. De Siena, and Q. Gan. Three-dimensional kernel-based coda attenuation imaging of caldera structures controlling the 1982-84 campi flegrei unrest. *Journal of Volcanology and Geothermal Research*, 381:273–283, 2019.

K. Aki. Attenuation of shear-waves in the lithosphere for frequencies from 0.05 to 25 Hz. *Physics of the Earth and Planetary Interiors*, 21:50–60, 1980.

K. Aki and B. Chouet. Origin of coda waves: Source, attenuation, and scattering effects. *Journal of Geophysical Research*, 80:3322–3342, 1975.

R. Aster, B. Borchers, and C. Thurber. *Parameter Estimation and Inverse Problem*, volume 90 of *International Geophysics Series, Advances in Geophysics*. Elsevier, Amsterdam, 2005.

L. V. Block. *Joint hypocenter–velocity inversion of local earthquake arrival time data in two geothermal regions*. Ph.d. dissertation, Massachusetts Institute of Technology, Cambridge, 1991.

M. Calvet and L. Margerin. Lapse time dependence of coda Q: anisotropic multiple-scattering models and application to Pyrenees. *Bulletin of the Seismological Society of America*, 103(3):1993–2010, 2013. doi: 10.1785/0120120239.

M. Calvet, M. Sylvander, L. Margerin, and A. Villaseñor. Spatial variations of seismic attenuation and heterogeneity in the Pyrenees: Coda Q and peak delay time analysis. *Tectonophysics*, 608:428–439, 2013.

V. F. Cormier and C. J. Sanborn. Trade-offs in parameters describing crustal heterogeneity and intrinsic attenuation from radiative transport modeling of high-frequency regional seismograms. *Bulletin of the Seismological Society of America*, 109(1):312–321, 2019.

L. De Siena, E. Del Pezzo, F. Bianco, and A. Tramelli. Multiple resolution seismic attenuation imaging at Mt. Vesuvius. *Physics of the Earth and Planetary Interiors*, 173:17–32, 2009.

L. De Siena, E. Del Pezzo, and F. Bianco. Campi Flegrei seismic attenuation image: evidences of gas resevoirs, hydrotermal basins and feeding systems. *Journal of Geophysical Research*, 115(B0): 9312–9329, 2010.

L. De Siena, E. Del Pezzo, C. Thomas, A. Curtis, and L. Margerin. Seismic energy envelopes in volcanic media: in need of boundary conditions. *Geophysical Journal International*, 192(1):326–345, 2013.

L. De Siena, C. Thomas, and R. Aster. Multi-scale reasonable attenuation tomography analysis (MuRAT): An imaging algorithm designed for volcanic regions. *Journal of Volcanology and Geothermal Research*, 277:22–35, 2014a.

L. De Siena, C. Thomas, G. P. Waite, S. C. Moran, and S. Klemme. Attenuation and scattering tomography of the deep plumbing system of Mount St. Helens. *Journal of Geophysical Research: Solid Earth*, 119(11):8223–8238, 2014b.

L. De Siena, M. Calvet, K. J. Watson, A. Jonkers, and C. Thomas. Seismic scattering and absorption mapping of debris flows, feeding paths, and tectonic units at Mount St. Helens volcano. *Earth and Planetary Science Letters*, 442:21—31, 2016.

L. De Siena, A. Amoruso, E. D. Pezzo, Z. Wakeford, M. Castellano, and L. Crescentini. Space-weighted seismic attenuation mapping of the aseismic source of Campi Flegrei 1983?1984 unrest. *Geophysical Research Letters*, pages n/a–n/a, 2017a. ISSN 1944-8007. doi: 10.1002/2017GL072507. URL `http://dx.doi.org/10.1002/2017GL072507`. 2017GL072507.

L. De Siena, G. Chiodini, G. Vilardo, E. Del Pezzo, M. Castellano, S. Colombelli, N. Tisato, and G. Ventura. Source and dynamics of a volcanic caldera unrest: Campi flegrei, 1983–84. *Scientific reports*, 7(1):1–13, 2017b.

E. Del Pezzo. Seismic wave scattering in volcanoes. In *Earth Heterogeneity and Scattering Effects of Seismic Waves*, volume 50 of *Advances in Geophysics*, chapter 13, pages 353–369. Elsevier, Amsterdam, 2008.

E. Del Pezzo and F. Bianco. Two-layer earth model corrections to the MLTWA estimates of intrinsic- and scattering-attenuation obtained in a uniform half-space. *Geophysical Journal International*, 182 (2):949–955, 2010.

E. Del Pezzo and J. M. Ibáñez. Seismic coda-waves imaging based on sensitivity kernels calculated using an heuristic approach. *Geosciences*, 10(8):304, 2020.

E. Del Pezzo, F. Bianco, and G. Saccorotti. Separation of intrinsic and scattering q for volcanic tremor: an application to etna and masaya volcanoes. *Geophysical Research Letters.*, 28(16):3083–3086, 2001.

E. Del Pezzo, F. Bianco, L. De Siena, and A. Zollo. Small scale shallow attenuation structure at Mt. Vesuvius. *Physics of the Earth and Planetary Interiors*, 157:257–268, 2006.

E. Del Pezzo, J. Ibañez, J. Prudencio, F. Bianco, and L. De Siena. Absorption and scattering 2-D volcano images from numerically calculated space-weighting functions. *Geophysical Journal International*, 206(2):742–756, 2016.

E. Del Pezzo, A. De La Torre, F. Bianco, J. Ibanez, S. Gabrielli, and L. De Siena. Numerically calculated 3d space-weighting functions to image crustal volcanic structures using diffuse coda waves. *Geosciences*, 8(5):175, 2018.

M. D. P. Di Martino, L. De Siena, D. Healy, and S. Vialle. Petro-mineralogical controls on coda attenuation in volcanic rock samples. *Geophysical Journal International*, 2021.

T. Eulenfeld and U. Wegler. Measurement of intrinsic and scattering attenuation of shear waves in two sedimentary basins and comparison to crystalline sites in germany. *Geophysical Journal International*, 205(2):744–757, 2016.

M. C. Fehler, P. Roberts, and T. Fairbanks. A temporal change in coda wave attenuation observed during an eruption of Mount St. Helens. *Journal of Geophysical Research*, 93:4367–4373, 1988.

M. C. Fehler, M. Hoshiba, H. Sato, and H. Obara. Separation of scattering and intrinsic attenuation for the Kanto-Tokai region, Japan, using mesurements of S-wave energy versus hypocentral distance. *Geophysical Journal International*, 108:787–800, 1992.

M. C. Fehler, H. Sato, and L. J. Huang. Envelope broadening of outgoing waves in 2D random media: a comparison between the Markov approximation and numerical simulations. *Bulletin of the Seismological Society of America*, 90:914–928, 2000.

S. Gabrielli, L. De Siena, F. Napolitano, and E. Del Pezzo. Understanding seismic path biases and magmatic activity at mount st helens volcano before its 2004 eruption. *Geophysical Journal International*, 222(1):169–188, 2020.

S. Gazzola, P. C. Hansen, and J. G. Nagy. Ir tools: a matlab package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81(3):773–811, 2019.

P. C. Hansen. Regularization tools version 4.0 for matlab 7.3. *Numerical algorithms*, 46(2):189–194, 2007.

P. Ho-Liu, H. Kanamori, and R. W. Clayton. Applications of attenuation tomography to imperial valley and coso-indian wells region, southern california. *Journal of Geophysical Research*, 93(B9): 10501–10520, 1988.

L. Margerin. Diffusion approximation with polarization and resonance effects for the modelling of seismic waves in strongly scattering small-scale media. *Geophysical Journal International*, 192(1): 326–345, 2013.

L. Margerin. Breakdown of equipartition in diffuse fields caused by energy leakage. *The European Physical Journal Special Topics*, 226(7):1353–1370, 2017.

K. Mayeda, S. Koyanagi, M. Hoshiba, K. Aki, and Y. Zeng. A comparative study of scattering, intrinsic and coda q for hawaii, long valley, and central california between 1.5 and 15.0 hz. *Journal of Geophysical Research*, 97:6643–6659, 1992.

J. Mayor, L. Margerin, and M. Calvet. Sensitivity of coda waves to spatial variations of absorption and scattering: radiative transfer theory and 2-d examples. *Geophysical Journal International*, page ggu046, 2014.

J. Mayor, M. Calvet, L. Margerin, O. Vanderhaeghe, and P. Traversa. Crustal structure of the Alps as seen by attenuation tomography. *Earth and Planetary Science Letters*, 439:71–80, 2016.

I. B. Morozov. Geometrical attenuation, frequency dependence of q, and the absorption band problem. *Geophysical Journal International*, 175(1):239–252, 2008.

I. B. Morozov. Mechanisms of geometrical seismic attenuation. *Annals of Geophysics*, 54(3), 2011.

F. Napolitano, L. De Siena, A. Gervasi, I. Guerra, R. Scarpa, and M. La Rocca. Scattering and absorption imaging of a highly fractured fluid-filled seismogenetic volume in a region of slow deformation. *Geoscience Frontiers*, 11(3):989–998, 2020.

C. Nardoni, L. De Siena, F. Cammarano, F. Magrini, and E. Mattei. Modelling regional-scale attenuation across italy and the tyrrhenian sea. *Physics of the Earth and Planetary Interiors*, page 106764, 2021.

K. Nishigami. A new invertion method of coda waveforms to determine spatial distribution of coda scatterers in the crust and uppermost mantle. *Geophysical Research Letters*, 12(18):2225–2228, 1991.

A. Obermann, T. Planes, E. Larose, C. Sens-Schönfelder, and M. Campillo. Depth sensitivity of seismic coda waves to velocity perturbations in an elastic heterogeneous medium. *Geophysical Journal International*, 194(1):372–382, 2013.

J. C. L. Paasschens. Solution of the time-dependent Boltzmann equation. *Physical Review*, 56(1): 1135–1141, 1997.

C. Pacheco and R. Snieder. Time-lapse travel time change of multiply scattered acoustic waves. *The Journal of the Acoustical Society of America*, 118(3):1300–1310, 2005.

A. Pisconti, E. Del Pezzo, F. Bianco, and S. de Lorenzo. Seismic q estimates in umbria marche (central italy): hints for the retrieval of a new attenuation law for seismic risk. *Geophysical Journal International*, 201(3):1370–1382, 2015.

J. Prudencio, J. M. Ibáñez, A. García-Yeguas, E. Del Pezzo, and A. M. Posadas. Spatial distribution of intrinsic and scattering seismic attenuation in active volcanic islands–ii: Deception island images. *Geophysical Journal International*, 195(3):1957–1969, 2013.

J. Prudencio, L. De Siena, J. Ibáñez, E. Del Pezzo, A. García-Yeguas, and A. Díaz-Moreno. The 3D Attenuation Structure of Deception Island (Antarctica). *Surveys in Geophysics*, 36(3):371–390, 2015.

N. Rawlinson and W. Spakman. On the use of sensitivity tests in seismic tomography. *Geophysical Journal International*, 205(2):1221–1243, 2016.

T. Saito, H. Sato, and M. Ohtake. Envelope broadening of spherically outgoing waves in three-dimensional random media having power law spectra. *Journal of Geophysical Research*, 107(B5): 2089–2103, 2002.

C. J. Sanborn, V. F. Cormier, and M. Fitzpatrick. Combined effects of deterministic and statistical structure on high-frequency regional seismograms. *Geophysical Journal International*, 210(2):1143–1159, 2017.

H. Sato. Broadening of seismogram envelopes in the randomly inhomogeneous lithosphere based on the parabolic approximation: southeastern Honshu, Japan. *Journal of Geophysical Research*, 94: 17735–17747, 1989.

H. Sato. Envelope broadening and scattering attenuation of a scalar wavelet in random media having power-law spectra. *Geophysical Journal International*, 204(1):386–398, 2016.

H. Sato, M. C. Fehler, and T. Maeda. *Seismic Wave Propagation and Scattering in the heterogeneous Earth: Second Edition*. Springer, New York, USA, 2012.

P. Sketsiou, F. Napolitano, A. Zenonos, and L. De Siena. New insights into seismic absorption imaging. *Physics of the Earth and Planetary Interiors*, 298:106337, 2020.

P. Sketsiou, L. De Siena, S. Gabrielli, and F. Napolitano. 3-d attenuation image of fluid storage and tectonic interactions across the pollino fault network. *Geophysical Journal International*, 226(1): 536–547, 2021.

T. Takahashi, H. Sato, T. Nishimura, and K. Obara. Strong inhomogeneity beneath Quaternary volcanoes revealed from the peak delay analysis of S-wave seismograms of microearthquakes in northeastern Japan. *Geophysical Journal International*, 168(1):90–99, 2007.

T. Takahashi, H. Sato, T. Nishimura, and K. Obara. Tomographic inversion of the peak delay times to reveal random velocity fluctuations in the lithosphere: method and application to northeastern japan. *Geophysical Journal International*, 178(3):1437–1455, 2009.

S. Takemura, T. Furumura, and T. Saito. Distortion of the apparent s-wave radiation pattern in the high-frequency wavefield: Tottori-ken seibu, japan, earthquake of 2000. *Geophysical Journal International*, 178(2):950–961, 2009.

J. N. Tripathi, M. Sato, and M. Yamamoto. Envelope broadening characteristics of crustal earthquakes in northeastern Honshu, Japan. *Geophysical Journal International*, 182(2):988–1000, 2010.

M. van Laaten, T. Eulenfeld, and U. Wegler. Comparison of multiple lapse time window analysis and qopen to determine intrinsic and scattering attenuation. *Geophysical Journal International*, 228(2): 913–926, 2021.

U. Wegler. Analysis of Multiple Scattering at Vesuvius Volcano, Italy, using Data of the TomoVes active seismic experiment. *Journal of Volcanology and Geothermal Research*, 128:45–63, 2003.

U. Wegler. Diffusion of seismic waves in layered media: boundary conditions and analytic solutions. *Geophysical Journal International*, 163:1123–1135, 2005.

U. Wegler and B. G. Luehr. Scattering behaviour at Merapi Volcano (Java) revealed from an active seismic experiment. *Geophysical Journal International*, 145:579–592, 2001.

K. Yoshimoto, H. Sato, and M. Ohtake. Frequency-Dependent Attenuation of P and S Waves in the Kanto Area, Japan, based on the Coda-Normalization Method. *Geophysical Journal International*, 114:165–174, 1993.