Scaling Full Seismic Waveform Inversions

Lion Krischer

München 2017

Scaling Full Seismic Waveform Inversions

Lion Krischer

Dissertation
zur Erlangung des Doktorgrades
an der Fakultät für Geowissenschaften der
Ludwig-Maximilians-Universität München

vorgelegt von
Lion Krischer
aus München

München, den 04.04.2017

Erstgutachter: Prof. Dr. Heiner Igel
Zweitgutachter: Prof. Dr. Andreas Fichtner
Tag der mündlichen Prüfung: 19. Juli 2017

Für Lisa, Joschua, Len und Nolan.

Summary

The main goal of this research study is to scale full seismic waveform inversions using the adjoint-state method to the data volumes that are nowadays available in seismology. Practical issues hinder the routine application of this, to a certain extent theoretically well understood, method. To a large part this comes down to outdated or flat out missing tools and ways to automate the highly iterative procedure in a reliable way.

This thesis tackles these issues in three successive stages. It first introduces a modern and properly designed data processing framework sitting at the very core of all the consecutive developments. The ObsPy toolkit is a Python library providing a bridge for seismology into the scientific Python ecosystem and bestowing seismologists with effortless I/O and a powerful signal processing library, amongst other things.

The following chapter deals with a framework designed to handle the specific data management and organization issues arising in full seismic waveform inversions, the Large-scale Seismic Inversion Framework. It has been created to orchestrate the various pieces of data accruing in the course of an iterative waveform inversion.

Then, the Adaptable Seismic Data Format, a new, self-describing, and scalable data format for seismology is introduced along with the rationale why it is needed for full waveform inversions in particular and seismology in general.

Finally, these developments are put into service to construct a novel full seismic waveform inversion model for elastic subsurface structure beneath the North American continent and the Northern Atlantic well into Europe. The spectral element method is used for the forward and adjoint simulations coupled with windowed time-frequency phase misfit measurements. Later iterations use 72 events, all happening after the USArray project has commenced, resulting in approximately 150‘000 three components recordings that are inverted for. 20 L-BFGS iterations yield a model that can produce complete seismograms at a period range between 30 and 120 seconds while comparing favorably to observed data.

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

1.1 A Curious Tale

Mankind has always been fascinated with what lies beneath our very feet. Folklore and early religions told of lands deep inside our planet that were strongly interweaved with the peoples' beliefs and fears. Mythological places like the Christian hell, the Greek underworld governed by Hades, or the Hindu kingdom of Shambhala carry meaning with most people even nowadays, be it through education or popular culture in general. As one cannot simply dig a deep hole and look into the insides of Earth, this turned out to be surprisingly hard to disprove. Serious scientists like Edmond Halley, famous for computing the orbit of Halley's Comet, put forth fairly plausible theories regarding a hollow Earth and others like John Cleaves Symmes, Jr. based their entire careers on them; figure 1.1 shows their conjectures. It took centuries of theoretical and observational discoveries from astro- and geophysics to collect enough evidence to confidently refute these theories and until today our most detailed pictures of the inner Earth originate in geophysics, in particular in seismology.

The beginning of the 20th century saw a slew of insights that sparked an increase of knowledge about Earth's structure. The first wave until the 1930s discovered the first-order features of our planet: Zoeppritz and co-authors compiled accurate travel time tables and assembled them into one dimensional Earth models (Zoeppritz, 1907; Zoeppritz et al., 1912), Mohorovičić discovered his discontinuity (Mohorovičić, 1910) in 1910, followed by the determination of the radius of Earth's outer core by Gutenberg (1913). The fluid nature of the outer core was deduced with various arguments by Jeffreys (1926), in the same year that Gutenberg discovered the asthenosphere (Gutenberg, 1926) and the final piece of the puzzle, the inner core, was contributed by Inge Lehmann (Lehmann, 1936). [1]

After the recognition of the basic features of Earth's structure, seismology determined averaging 1-D models starting with the famous travel time tables by Jeffreys and Bullen (1940) and the associated velocity model, which even nowadays hold up remarkably well. Refined 1-D Earth models have been developed over the years (Dziewonski and Anderson, 1981; Kennett and Engdahl, 1991; Kennett et al., 1995) but the basic features are not radically different.

Recognizing the laterally heterogeneous nature of Earth, the first ray-theoretical imaging procedures were carried out in the 1970s (Aki et al., 1977). Computational as well as methodological and technical developments lead to ever more complex and sophisticated models requiring more and more data and computational power (Kissling, 1988; Spakman, 1991; Dahlen et al., 2000; Rawlinson and Sambridge, 2003; Friederich, 2003). The current state-of-the-art are so-called full seismic waveform inversions that attempt to exploit as much information from seismic waveforms as physically feasible. They are the central topic of this thesis.

[1] This simplified view omits many details and important steps in-between, please see Gutenberg, Beno (1959), Dziewonski and Romanowicz (2007), and references therein for more detailed accounts.

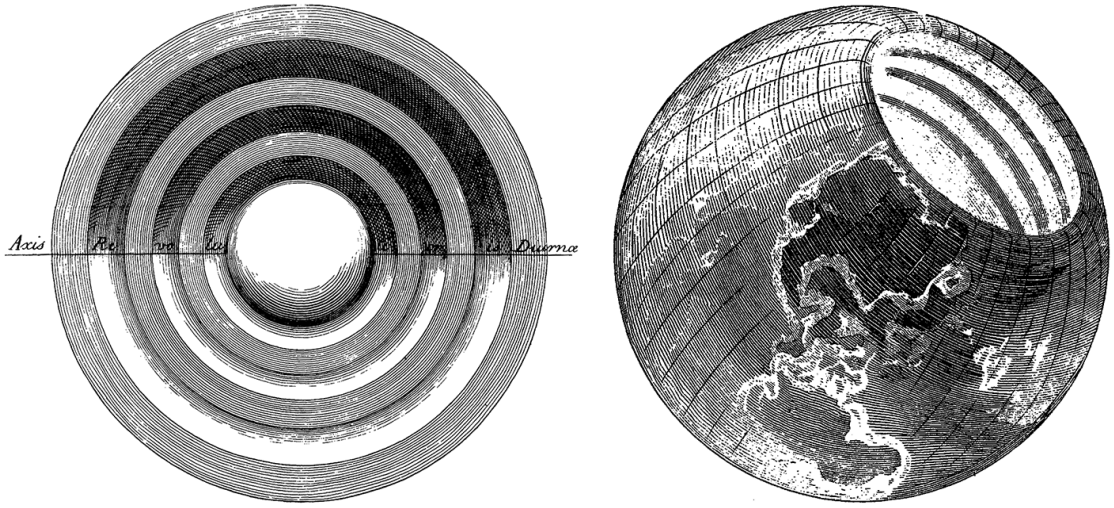

Figure 1.1.: Two to some extent scientifically rigorous models of a hollow Earth. The left illustration (modified after Halley, 1686) shows the work of Edmond Halley who devised a planet with three concentric shells and an inner core. Each shell has its own magnetic poles and rotation speed which he used to explain certain compass readings. The right side (modified after Symmes, 1780–1829) shows a model that was formulated about a century later by John Cleaves Symmes, Jr. with four inner shells and wide openings at the poles. Both models had decent support at the time.

The period from the 1980s until today has been dubbed the era of tomography and broadband digital seismic networks by Dziewonski and Romanowicz (2007). It brought along the deployment of the Global Seismographic Network (GSN), a global network of high-quality broadband seismic instruments which in turn lead to the formation of the Federation of Digital Seismographic Networks (FDSN; Romanowicz and Dziewonski, 1986) in order to coordinate these efforts. Today, it is still the main body to globally govern and unify for example network codes and data exchange formats. Regional networks keep increasing their station code and large-scale projects like the USArray project (`http://www.usarray.org`), AlpArray (`http://www.alparray.ethz.ch`, Fuchs et al. (2016), Molinari et al. (2016), Govoni et al. (2017)), and ChinaArray (`http://chinaarray.org/`) present the current culmination of that trend and the total amount of available data is only expected to increase in the future.[2]

That particular combination of theoretical progress, computational advances, and constantly increasing availability of high-quality data enables the construction of more realistic and better Earth model. In order to do this, our community needs to develop and adapt new techniques to process, manage, and store this data and to tie it up in more scalable and cohesive workflows. This thesis describes the rationale and motivations behind some of these development for seismology in general and with respect to full seismic waveform inversions in particular, before applying them to derive a new continental-scale full seismic waveform model.

[2] All URLs in the introduction last accessed March 2017.

1.2 Objectives and Outline

Imaging the subsurface structure of Earth is one of seismology's paramount goals. Globally distributed earthquakes excite seismic waves that travel through the globe. Deviations from the expected structure influence the passing seismic waves and these differences enable seismologists to map Earth's structure. Full seismic waveform inversions do that by comparing large parts of complete seismograms with theoretical predictions in the form of synthetic waveforms. The basic theory is, to a certain part, well understood (see e.g. Tarantola, 1988; Tromp et al., 2005; Fichtner et al., 2006, 2009; Tape et al., 2010; Zhu et al., 2012; Liu and Gu, 2012; Bozdağ et al., 2016) but practical issues hinder these techniques to be scaled in a reliable way to the data volumes in existence. This thesis aims at solving those: The first three chapters each tackle one particular problem and the final chapter synthesizes these techniques to develop a new seismic velocity model for structure beneath Northern America and the Northern Atlantic.

Chapter 2 introduces the ObsPy toolkit, a modern I/O and data processing library for seismology. It forms the foundation of all the following developments by offering well tested and engineered routines to read, write, download, and process data in a powerful and general purpose programming language.

Chapter 3 shows the LASIF framework which builds on the foundation of ObsPy to manage and organize the data in large-scale full seismic waveform inversions. It aims at orchestrating the various types of data accruing in waveform inversions in a clean and reproducible manner in order to reduce the time to research.

Chapter 4 introduces the Adaptable Seismic Data Format, a new format for seismological data which is developed in response to issues discovered when performing large waveform inversions. It is applicable to all branches of seismology and enables the construction of complete and independent data sets that can be exchanged with others.

Chapter 5 finally uses these developments for a new large-scale full seismic waveform model for Northern America and the Northern Atlantic. The explicit goal of this inversion was to harness as much data as possible and physically reasonable in order to test and validate the previous developments.

Chapers A and *B* do not really follow the central theme of this thesis but are nonetheless presented here as the work was carried out during the preparation of this thesis. They introduce Instaseis, a package to quickly extract accurate broadband high-frequency synthetic seismograms from a 1-D velocity model, and the related Syngine service now running at the IRIS DMC.

Modified version of chapters *2*, *3*, *4*, *A*, and *B* have been published in various peer reviewed journals which are mentioned at the beginning of each chapter, a publication for the contents of chapter *5* is in preparation.

1.3 Software

A large part of this thesis deals with the development of software libraries, applications, and tools. While this is not necessarily exactly aligned with science in the traditional sense, I do

consider them a significant part of my contributions to the seismological community and crucial ingredient of my research.

A key driver is the development of generically useful utilities that are not just one-off solutions but can be used by others to facilitate and enable their research. The ObsPy software and the ASDF data format are generically useful for all kinds of seismological problems, LASIF is specific to full waveform inversions but not tied to any particular tomography. Thus, these tools can be and are used by others in turn enabling new discoveries and pushing the whole field forward.

As software constantly evolves it makes little sense to show source code and extensive documentation here but instead I refer to my Github page (`https://github.com/krischer/`) where interested readers can find links and more information.

2
Processing Data with ObsPy

The core of this work deals with managing and processing large amounts of data and at the very heart of that lies the ObsPy library. The work presented in all the following chapters is in some way based on it. It is a Python library that can read, write, acquire, and process all kinds of seismological data and by now serves as a foundational layer for many tools and applications in the seismological community. The effort is spearheaded and organized by the seismology group in at the LMU in Munich.

The chapter at hand details some of the abstraction within ObsPy and how it utilizes a combination of the scientific Python stack and bindings to existing and established libraries to provide a powerful and easy-to-use Python interface for seismologists. This chapter has been published in slightly altered form in:

> *Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., & Wassermann, J.* (2015).
> **ObsPy: a bridge for seismology into the scientific Python ecosystem.**
> Computational Science & Discovery, 8(1), 14003–14020.
> `https://doi.org/10.1088/1749-4699/8/1/014003`

Further details are published in Beyreuther et al. (2010) and Megies et al. (2011); examples, use cases, and applications are collected at `https://obspy.org` (last accessed March 2017).

2.1 Introduction

Python is an interpreted, general purpose programming language that becomes a powerful language for numerical analysis when coupled with the NumPy (Oliphant, 2007) and SciPy (Jones et al., 2001) packages. The former provides an efficient array interface, while the latter contributes all kinds of scientific functionality from advanced linear algebra routines, over optimization routines, to sparse matrix methods, and a lot more. These two, together with the plotting library matplotlib (Hunter, 2007), make up the core of the so-called SciPy stack which is well suited for a large variety of applications.

We developed ObsPy, a Python library for seismology intended to facilitate the development of seismological software packages and workflows, to utilize these abilities and provide a bridge for seismology into the larger scientific Python ecosystem.

Scientists in many domains who wish to convert their existing tools and applications and take advantage of a platform like the one Python provides are confronted with several hurdles like special file formats, unknown terminology, and no suitable replacement for a non-trivial piece of software.

We present an approach to implement a domain specific time series library on top of the scientific SciPy stack. In doing so, we show a realization of an abstract internal representation of time series data permitting I/O support for a diverse collection of file formats. Then we detail the integration and repurposing of well established legacy codes enabling them to be used in modern workflows composed in Python. In the last part of this chapter we present a case study on how to integrate research codes into ObsPy opening them to the broader community.

While the implementations presented here are specific to seismology many of the described concepts and abstractions are directly applicable to other sciences, especially to those with an emphasis on time series analysis and data handling.

The scientific Python ecosystem offers a wealth of possibilities for all fields of science, mathematics, and engineering. This enables the creation of versatile workflows and applications. Over the years many areas of science developed their own set of file formats, tools and analysis software adapted to suit their particular needs. Extending these to be more flexible often ends up being hard to impossible. A more general approach like the one offered by the SciPy stack is desirable. An added advantage is that many high-quality scientific libraries that are built on top of this stack are in existence which can readily be used to assemble full workflows.

ObsPy (Beyreuther et al., 2010) provides read and write support for essentially all file formats commonly distributed within the seismological community superseding an abundance of file format converters. Seismology usually distinguishes three types of data: waveform data representing the actual time series, station meta data providing information about the seismic receivers, and event meta data representing sources of seismic waves of either anthropogenic or natural origins. All of these can be read, in various manifestations, with ObsPy, but this work focusses on time-series waveform data. On top of this broad I/O support it offers signal processing routines utilizing a jargon prevailing amongst seismologists. A third milestone is the integrated access to data distributed by a wide range of seismic data centers worldwide and finally it integrates a number of existing special purpose libraries in use in seismology while unifying all functionality with an easy to use interface.

2.2 Domain Specific Time Series Analysis

The ObsPy library contains a domain-specific time series analysis toolkit which enables seismologist to construct processing workflows in a notation familiar to them. It thus provides an interface to the vast functionality offered by NumPy and SciPy to domain experts which might otherwise hesitate to invest into learning Python and its scientific ecosystem. This section contains a technical description of how a number of different time series file formats can be handled in a unified manner and how to use a plug-in system to map general signal processing routines provided by NumPy and SciPy to the aforementioned convenient interface. This approach has proven itself to work very well and we believe it can easily be applied to other fields.

2.2.1 Seismic Waveforms

Seismology studies the propagation of elastic waves through mostly solid media. The seismic waves result from natural (tectonic earthquakes, volcanoes, ocean waves, …) or man-made (nuclear explosions, quarry blasts, induced earthquakes, …) sources and are measured and recorded at seismographs. These perform point measurements of the elastic wavefield in up to three orthogonal directions. Each direction or component measures either the displacement, velocity, or acceleration of the ground motion in form of a one-dimensional, time-dependent signal. The resulting equally sampled time series are called seismograms or seismic waveforms.

With the emergence of digital seismology in the last couple of decades numerous seismic waveform file formats surfaced. Some, like the MiniSEED format (Incorporated Research Institutions for Seismology (IRIS), 2012) for data archiving and streaming, were designed with a specific purpose in mind while the majority of formats are by-products of seismic signal processing or analysis packages that used custom formats for their I/O. The adoption of some of these software packages caused their I/O formats to become widely used data exchange formats - a purpose they were not designed for. A recurring problem for example is the byte ordering: most formats do not specify whether they are written in big or little endian and for many formats examples of both can be found. This myriad of file formats and software suites only accepting one particular file format caused many format converters to be written and distributed.

Each available waveform format in essence stores one or more time series and some meta-information about it. The essential ingredients of these formats can be distilled to a common subset of information enabling the use of a unified internal representation. ObsPy provides an implementation of that idea which is described in the following.

2.2.2 Characteristics of Waveform Data Formats

Assuming that a seismic data recording starts at time A, ends at time B and is equidistantly sampled without gaps and overlaps it can be uniquely described by the time of the first sample, the sampling rate, the instrument it was recorded with, and an array containing the signal. This is the common core information and all waveform file formats have to contain it in some fashion. In seismology receivers are uniquely identified by the so called SEED identifiers Incorporated Research Institutions for Seismology (IRIS) (2012) consisting of four short strings. A globally

coordinated effort attempts to assure that this system stays intact and as a direct result most waveform formats adhere to it, offering a solution to the receiver location issue.

Seismic data also oftentimes has gaps and overlaps meaning that the data is not evenly sampled and the previous assumption does not hold true anymore. Gaps can result from short losses of power at the recording station or problems with the data transfer. Clock drifts and corrections can be one of the reason for the overlap of data points. Some data formats and ObsPy deal with this by allowing the storage of multiple chunks of waveform data each being equally sampled and generally well behaved in itself.

Any additional information a waveform file format might support within ObsPy is stored separately as detailed in the next section.

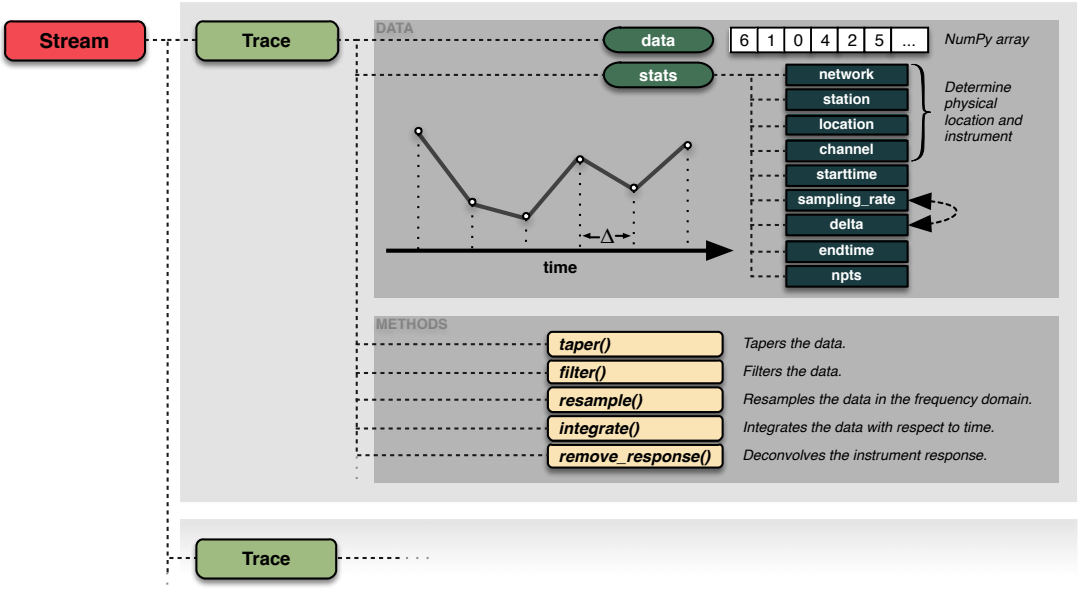

Figure 2.1.: Illustration of ObsPy's internal waveform data representation in form of the `Trace` objects gathered in a `Stream` object. Each `Trace` object represents a contiguous, equally sampled time series with the samples stored in a *NumPy* array, additional metadata is placed under the `stats` attribute. The `Stats` object will keep the meta-information self-consistent by making use of custom item setters. A collection of domain specific signal processing methods enables scientists to construct workflows in a terminology they are familiar with.

2.2.3 Internal Data Representation

Within ObsPy waveform data is represented by a `Stream` object which acts as a container for any number of `Trace` objects. ObsPy defines a `Trace` to contain a single, contiguous, equally sampled time window of waveform data alongside the necessary meta-information. Each `Trace` object has a `data` attribute, which is a one dimensional *NumPy* array. All further information is located in the dictionary-like `stats` attribute.

The `stats` object stores the four SEED identifiers `network`, `station`, `location`, and `channel` denoting the recording's physical location and instrument. It furthermore contains the times of the first and last sample, the data's sampling rate and interval, and the number of samples. This information is partially redundant and ObsPy takes care that it stays consistent. For example the `endtime` attribute is read-only and will be adjusted automatically if the start time,

the sampling rate, or the number of samples in the array has changed. Any additional information a specific file format might contain not covered by this abstraction will be stored in a container inside the `stats` object. See figure 2.1 for an illustration of the described internal representation.

2.2.4 Broad Data Format Support by Means of a Plug-In System

In order to support as many waveform file formats as feasible a modular plug-in approach has been chosen with each supported data format being its own submodule and registering with ObsPy with the help of *pkg-util's* plug-in system. Listing 2.1 shows the registration of an example waveform plug-in. A file format submodule has to implement two functions with specified interfaces, a third one is optional:

- `is_format(filename)`: Format identification function. Returns `True` if the passed file is of the submodule's format, `False` otherwise.
- `read_format(filename, **kwargs)`: Reads the file and returns a `Stream` object containing a representation of the file's data.
- `write_format(stream, filename, **kwargs)`: Writes a `Stream` object to the given filename. This function is optional, if not given no write support for the format will be available.

```
ENTRY_POINTS = {
    ...
    "obspy.plugin.waveform": [
        ...
        "MSEED = obspy.mseed.core",
        ...
    ]

    ...

    "obspy.plugin.waveform.MSEED": [
        "isFormat = obspy.mseed.core:isMSEED",
        "readFormat = obspy.mseed.core:readMSEED",
        "writeFormat = obspy.mseed.core:writeMSEED",
    ],
    ...
}
```

Listing 2.1.: Excerpt from ObsPy's **setup.py** script demonstrating how the waveform plug-in entry points are defined on the example of the MiniSEED plugin. *distutils* will then take care of registering the plug-in upon installation. Other Python modules can register their own waveform format plug-ins for rarely used formats.

The forced modularity of this strategy fosters a clean separation of concerns with each submodule being implemented and tested independently. In addition it allows users to extend ObsPy with I/O support for formats that are not used widely enough to justify integration into

the main ObsPy library. A common example for this are output formats of numerical waveform solvers. *pkg-utils'* plug-in system will register these additional formats for a seamless integration with the rest of ObsPy.

2.2.5 Format Autodetection and Usage

ObsPy comes with a top-level `read()` function, a single entry point when reading waveform data. See listing 2.2 for a usage example.

```
>>> import obspy
>>> st = obspy.read("filename")
>>> st
<obspy.core.stream.Stream at 0x2f284d0>
>>> print st
3 Trace(s) in Stream:
BW.RJOB..EHZ | 2009-08-24T00:20:03.000000Z - ... | 100.0 Hz, 3000 samples
BW.RJOB..EHN | 2009-08-24T00:20:03.000000Z - ... | 100.0 Hz, 3000 samples
BW.RJOB..EHE | 2009-08-24T00:20:03.000000Z - ... | 100.0 Hz, 3000 samples
```

Listing 2.2.: Snapshot of an interactive Python session demonstrating the usage of ObsPy's `read()` function. It will detect the file's format and call the appropriate reading routine. In case the `read()` routine detects a valid HTTP URL it will download the resource before proceeding. In case it detects an archive format it will be decompressed first.

The `read(filename, **kwargs)` routine calls all registered formats' `is_fileformat()` functions until one returns `True`. Depending on the format in question the `is_fileformat()` routine parses the first couple of bytes or performs more complicated heuristics. For performance reasons the format detection routines have to be fast. On top of that, ObsPy performs the format detection according to a manually curated list so that the most commonly used ones are tested first improving average performance. After the format has been determined the appropriate format's `read_fileformat()` function will be called which parses the file and returns a `Stream` object. The format detection can also be skipped by providing the format to the `read()` routine.

This structure permits the sharing of capabilities amongst all formats by implementing them as part of the `read()` routine. Examples of this are automatic file downloading if a valid HTTP URL is detected and the decompression of a number of different archive formats.

An important source for waveform data are web services provided by data centers around the globe. ObsPy implements clients able to interact with a comprehensive selection of those Megies et al. (2011). A waveform data request will end up being stored in a `Stream` object so the workflow following the data acquisition is identical no matter the origin of the data.

Stream Methods	
`merge()`	Attempts to merge `Trace` objects with the same ID.
`rotate()`	Rotates two or three-component `Stream` objects.
`select()`	Returns a new `Stream` with `Traces` matching the selection.

Trace Methods	
`decimate()`	Decimates the data by an integer factor.
`detrend()`	Removes a linear trend from the data.
`differentiate()`	Differentiates the data with respect to time.
`filter()`	Filters the data.
`integrate()`	Integrates the data with respect to time.
`normalize()`	Normalizes the data to its absolute maximum.
`remove_response()`	Deconvolves the instrument response.
`resample()`	Resamples the data in the frequency domain.
`taper()`	Tapers the data.
`trigger()`	Runs a triggering algorithm on the data.
`trim()`	Cuts the data to given start and end time.

Table 2.1.: Selection of processing methods for the `Stream` and `Trace` objects. Most `Trace` methods are also available on the `Stream` objects which will apply the chosen method to all its children.

2.2.6 Domain Specific Convenience Methods

A unified internal data representation opens the possibility to define methods transforming it. With respect to this ObsPy offers a comprehensive collection of signal processing routines frequently used in seismology by relying heavily on functionality coming with NumPy and SciPy. These are implemented to match the needs seismologists have of a certain processing operation while keeping the data self-consistent. An example of this is the `Trace.decimate()` method which will apply a filter, decimate the data, and adjust the sampling rate metadata of the times series. Caused by the potentially large number of samples, most operations are implemented as in-place modifications of the `Stream` and `Trace` objects.

Most processing methods are defined on a single `Trace`, the same methods on the `Stream` objects will call the corresponding method on each of its `Trace` children. This allows for a natural handling of three and more component data. A number of methods are specific to `Stream` methods. Consult table 2.1 for a selection of available signal processing routines.

In order to offer a large variety of functionality, ObsPy once again employs a plug-in system. Listings 2.3, 2.4, and 2.5 demonstrate this by example with the `Trace.taper()` method. The employed plug-in system empowers the usage of functionality defined by different modules into a simple domain specific API usable by scientists. It furthermore encourages code de-duplication and reuse and thus enables ObsPy to offer a large variety of different functions without having to implement and test the details in many cases.

```
...
"obspy.plugin.taper": [
  "cosine = obspy.signal.invsim:cosTaper",
  "barthann = scipy.signal:barthann",
  ...
]
...
```

Listing 2.3.: Excerpt from ObsPy's **setup.py** file. Upon installation these entry points will be registered and made available to *pkg-utils*. The code snippet illustrates this by registering different taper windows. At the time of writing ObsPy exposes 18 different taper windows mainly from the *scipy.signal* module. Usage is demonstrated in listing 2.5.

```
...
# retrieve function call from entry points
func = _getFunctionFromEntryPoint(
    "taper", type)
...
taper_sides = func(2 * wlen, **kwargs)
...
```

Listing 2.4.: Code fragments from within the `Trace` object's `taper()` method. It shows how the functions registered as illustrated in listing 2.3 are called to obtain different taper windows and how optional keyword arguments are passed to it. Usage is demonstrated in listing 2.5.

```
import obspy

st = obspy.read("filename")
# Taper with a cosine window.
st2 = st.copy().taper("cosine")
# Taper a with modified Bartlett-Hann window.
st3 = st.copy().taper("barthann")

# The methods can be chained for a compact notation.
st4 = st.copy().detrend("linear").taper("cosine")
```

Listing 2.5.: Usage of the functions registered and called in listings 2.3 and 2.4. Note how the actual taper windows can be and in this case are defined in different Python modules. On top of that this listing demonstrates method chaining which is possible because the methods return their object. The calls to `copy()` are necessary as most methods work in-place and therefore would modify the object. This memory-saving behavior is wanted in most cases and a concious design choice to match the average use case.

2.3 Integration of Legacy Codes

For certain tasks the seismological community frequently relies on codes that have been in use for a decade and more. This heavy exposure and application to a large variety of problems assures that they work as expected in almost all cases as most issues have already been discovered and fixed. While these codes are often cumbersome to use, it is desirable to keep them vivid and to profit from the investment made in their development in the past. ObsPy integrates a number of legacy codes, enabling the use of modern workflows and recent advances in data processing while simultaneously relying on stable and well tested code for a specific functionality. This section illustrates this on two examples.

2.3.1 Travel Time Calculations with iaspei-tau

The calculation of seismic travel times is a problem frequently encountered in seismology. If an earthquake occurs at point X the question is what seismic phases arrive at what time at point Y. The cost for full 3-D simulations of the whole earth (Tromp et al., 2008) are excessive for many applications and less accurate but faster approximations are often sufficient. In case a 1-D spherically symmetric earth model is applicable, the frequently used method of Buland and Chapman Buland and Chapman (1983) enables very fast travel time calculations. It has been implemented in a Fortran package commonly called *iaspei-tau* (Kennett and Engdahl, 1991; Snoke, 2009), a more feature-rich Java version of this algorithm is also available Crotwell et al. (1999). The *obspy.taup* submodule uses *ctypes* from the Python standard library to provide a pythonic interface to *iaspei-tau*, see listing 2.6. Figure 2.2 plots the travel times of an earthquake through the earth versus the geographic distance in degrees.

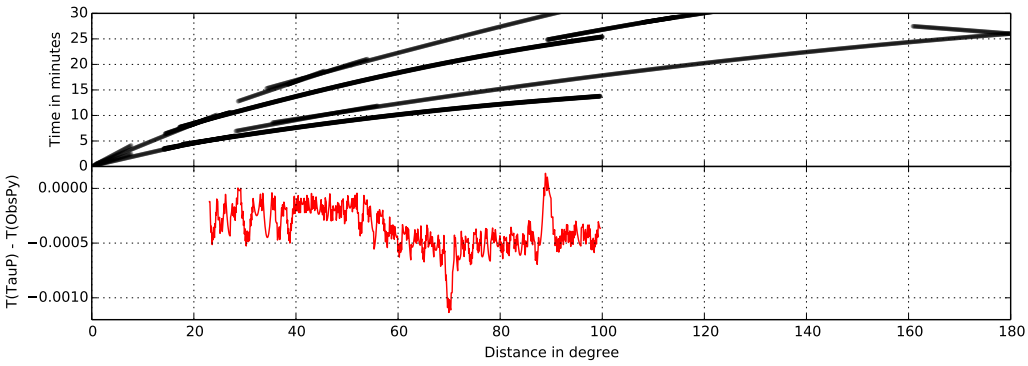

Figure 2.2.: Travel times of seismic waves through the 1-D earth model *ak135*. The top graph shows travel times for some seismic phases calculated by the *obspy.taup* module. The bottom plot shows the difference for the P phase travel times calculated with the *TauP Toolkit* (Crotwell et al., 1999) and *obspy.taup*. The deviations are mainly due to differing internal coordinate systems and are well understood by the community (Knapmeyer, 2005). The amplitude of the differences strongly depends on the used seismic phase and the source-receiver geometry.

```
>>> from obspy.taup import getTravelTimes
>>> getTravelTimes(delta=25.0, depth=10.0, model="ak135")
[{"phase_name": "P",
  "take-off angle": 28.375334,
  "time": 323.91006, ...},
 {"phase_name": "pP",
  "take-off angle": 151.61096,
  "time": 326.94455, ...},
 ...
```

Listing 2.6.: Usage of *obspy.taup* to attain travel times for seismic phases originating from a source with a distance of 25° and a depth of 10 kilometers. The distance `delta` is given in degrees assuming a spherical earth, the `depth` is in kilometers and the earth model is *ak135*. It returns a list instead of a dictionary as the phase names are non-unique for certain distance and phase combinations.

2.3.2 Instrument Responses: StationXML Harnesses evalresp

Seismic receivers distributed around the globe aim to measure and record ground motion as accurately as possible. Many factors influence the characteristics of the final waveform, amongst them the frequency response of the physical instrument, the effects of any amplifiers, of analog and digital filters, and of the digitalization. For many applications studying the earth it is crucial to eliminate these effects to get the best possible estimate of the true ground motion. The first step when correcting data for the influence of the seismic receiver and processing chain is to calculate the frequency response of the recording system. The seismograms are then deconvolved with this response to obtain a seismogram with physical units unbiased by instrumental effects in the frequency band of consideration. This process is known as instrument correction in seismology.

```
struct complex {
  double real;
  double imag;
};

struct pole_zeroType {
  int nzeros;
  int npoles;
  double a0;
  double a0_freq;
  struct complex *zeros;
  struct complex *poles;
};
```

Listing 2.7.: C struct representing the physical response of an instrument in form of poles and zeros. Recreating this with *ctypes* requires the definition of a Python class which is shown in listing 2.8.

```
import ctypes as C

...

class pole_zeroType(C.Structure):
    _fields_ = [
        ("nzeros", C.c_int),
        ("npoles", C.c_int),
        ("a0", C.c_double),
        ("a0_freq", C.c_double),
        ("zeros", C.POINTER(complex_number)),
        ("poles", C.POINTER(complex_number)),
    ]
```

Listing 2.8.: Python class representing the `pole_zeroType` C structure shown in listing 2.7 using the *ctypes* library. `complex_number` is already defined in this example.

Seismic recording systems are described by a linear chain of different stages or elements. The SEED data format Incorporated Research Institutions for Seismology (IRIS) (2012) accurately

represents these, is widely accepted by the community and has been in use since the early 90s. Recently, StationXML The International Federation of Digital Seismograph Networks (FDSN) (2014), the designated successor of SEED, has been developed and the community is starting to adopt it. Being an XML format it has many benefits in comparison to the binary SEED format regarding tool support, ease of data distribution and human readability. Except from some minor differences, both formats store the same information.

The standard workflow to calculate the frequency response of a seismic receiver system from SEED files involves *rdseed* IRIS DMC (2014b) to generate RESP files, a textual strict subset of SEED. These files are then fed into *evalresp* IRIS DMC (2014a) resulting in the frequency response. The detour via the RESP files is necessary as it is the only input file format *evalresp* accepts. Performing an instrument correction with metadata stored in StationXML files involves one additional step - the conversion of the StationXML files to SEED files using yet another tool.

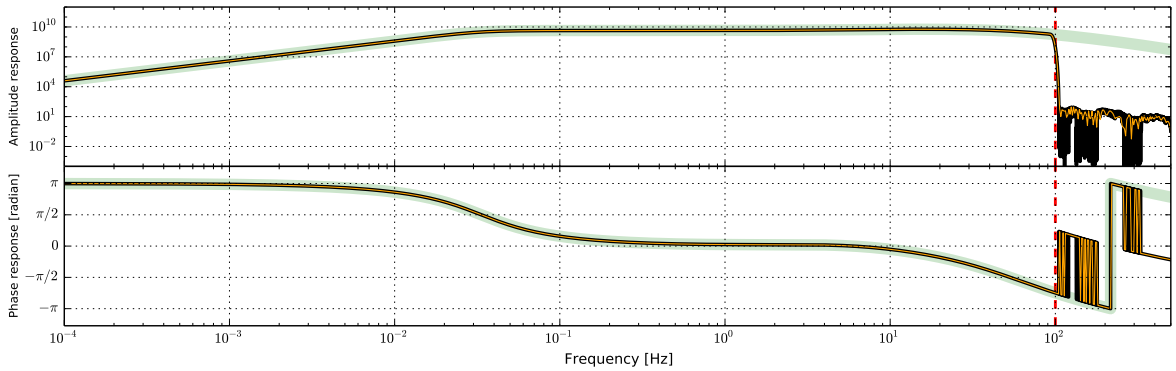

Figure 2.3.: Amplitude and phase response of channel XM.05..HHZ to convert a seismogram from meter per second to digital counts. The recording instrument is a Guralp CMG-6TD Seismometer. The green line shows the response if only the physical instrument and the gain from the analog to digital converter are taken into account, the black line also incorporates the three successive digital decimation stages, and the red line is the Nyquist frequency for this particular channel. The green and black lines have been calculated using ObsPy's integration of evalresp. For comparison the orange line shows the results for the same calculation performed by JEvalResp Instrumental Software Technologies, Inc. (2014), a Java port of evalresp and one of the very few software packages able to perform this calculation. The agreement is very good; the differences after the Nyquist frequency are due to the spacing of the discrete frequency values needed for plotting.

Conducting this in Python involves a number of system calls and unnecessary I/O operations making it ill suited to modern large data workflows. Replacing *evalresp* would be a major effort as many pitfalls exist when calculating instrument responses which can greatly influence the final result. Instead ObsPy offers a direct bridge from StationXML (parsed through *lxml*) to *evalresp* utilizing *ctypes* to call internal *evalresp* functions. This approach is not feasible for code that is still actively developed as no guarantees on the stability of the internal API are usually granted. However, *evalresp* is not in active development anymore and only receives an occasional maintenance commit.

ObsPy's internal representation of StationXML data is used to derive the nested C structures *evalresp*'s internal functions expect. These are defined and initialized using *ctypes* as demonstrated in listings 2.7 and 2.8. *Ctypes* also requires a definition of the function headers as shown in listings 2.9 and 2.10. Figure 2.3 displays the amplitude and phase response calculated from a StationXML file also illustrating the effect of different stages in the recording system.

A seismic receiver's recording system usually has several stages resulting in a fairly complex representation within *evalresp*. *Evalresp's* file parsing stage takes care of translating the SEED constructs to the appropriate internal structure and sometimes makes non-obvious decisions. To assure the correctness of the evalresp integration we performed an extensive test (Krischer, 2014). We define our integration of evalresp acting on StationXML files to be correct if the final response is equivalent to a response calculated by converting StationXML to SEED and SEED to RESP files on which *evalresp* is acting. We downloaded almost the complete set of StationXML inventory data available from IRIS, the largest data distributor in seismology. Data from over 27,000 stations, most with multiple recording channels defined for different periods in time result in well over 100,000 instrument responses. We iterated on our implementation until the tests passed giving confidence that our solution can be safely applied to any data encountered in the world wide community.

```
...

struct complex {
    double real;
    double imag;
};

...

void calc_resp(struct channel *chan,
    double *freq,
    int nfreqs,
    struct complex *output,
    char *out_units,
    int start_stage,
    int stop_stage,
    int useTotalSensitivityFlag);
```

Listing 2.9.: Function declaration in the C code of *evalresp*. This is the main function used to calculate the response. Note how the input takes a number of different parameters from simple integers to arrays of custom structs. The `complex` struct does not need to be defined on the Python side as the `numpy.complex128` dtype does have the same internal memory layout.

```
from obspy.signal.headers import clibevresp
import ctypes as C

...

clibevresp.calc_resp.argtypes = [
    C.POINTER(channel),
    np.ctypeslib.ndpointer(
        dtype='float64', ndim=1,
        flags='C_CONTIGUOUS'),
    C.c_int,
    np.ctypeslib.ndpointer(
        dtype='complex128', ndim=1,
        flags='C_CONTIGUOUS'),
    C.c_char_p, C.c_int, C.c_int,
    C.c_int]
clibevresp.calc_resp.restype = C.c_void_p
```

Listing 2.10.: The corresponding declaration to listing 2.9 in *ctypes*. The *numpy.ctypeslib* module provides array types which will result in automatic type, dimension, and flag checks upon function invocation. This results in a convenient calling syntax and error handling on the Python side before calling the shared library.

2.4 Integrating Research Code into ObsPy

2.4.1 Purpose

Contrary to the well-established, virtually bug-free legacy codes described above, many scientists using Python in their analytic workflow tend to develop bits and pieces of codes that are applied to their input data, and thus at first sight not useful for others. This general ascertainment is also true in seismology where input data can have different formats, outputs need to be compatible with different downstream processing software, and so forth. In this section, we first present a case study of the translation of a C code that can read an exotic format called

Win and then present a research code developed during a PhD thesis that is now integrated in ObsPy as a new module. Thanks to the pluggability of ObsPy, only the necessary processing steps need to be written.

2.4.2 Reading an Excotic Format - obspy.win

WIN is the format imagined by a Japanese company named Hakusan as default storage for their Datamark datalogger series. The data within a one minute WIN file is highly compressed, every second being compressed with a different level from the previous. Up to now, there were three ways to convert WIN to a more standard format, namely SAC: *Win2Sac* and *japan2sac* in Linux, *Win2Sac GUI* on Windows. *Win2Sac* codes come from Japan, while the japan2sac was written as part of a research project in Chambery (France) and is not completely finished. In order to process this data using state of the art routines and software, e.g. using ObsPy-based scripts, one has to convert the whole archive to SAC, which is readable by ObsPy. Each step of this process doubles the volume of the archives stored on disk, and potentially adds errors to the final product.

Thankfully, sources of *Win2Sac* are available online Ohmi (2014) and together with the data sheet of the format (winformat, 2014) describing the WIN format we have been able to write an ObsPy plug-in to read it. The final goal was to be able to read directly from the archive without duplication and process those data the same way as data originating from other file formats. Even if one's goal is to convert WIN to MiniSEED, the best solution today will be to write a Python script using ObsPy and no longer rely on a succession of steps.

2.4.3 Rewriting Research Code to an ObsPy Extension

The state of a volcano and its unrests can be evidenced by the amount of amplitude (or energy) recorded by seismic sensors on its flanks. RSAM (Real-time Seismic Amplitude Measurement (2.1)) was initially introduced by Endo and Murray (Endo and Murray, 1991) to forecast eruptions and assess the state of volcanic activity. One can also compute the standard deviation of the squared amplitudes, or RSEM (Real-time Seismic Energy Measurement (2.2)) De la Cruz-Reyna and Reyes-Dávila (2001) of the signals over a certain time interval divided by the number of measurements.

$$RSAM = \frac{1}{N}\sum_{i=1}^{n}(|A_i|) \tag{2.1}$$

$$RSEM = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(A_i - A_{avg})^2} \tag{2.2}$$

where A_i is the seismic signal's amplitude, A_{avg} is the average over N samples.

As multiple sources are overlapping, volcano-seismologists typically calculate the RSEM or RSAM values for separate, previously defined frequency bands (SSEM (Spectral Seismic Energy Measurement) Tárraga et al. (2006), similar to SSAM introduced by Stephens Stephens et al. (1994)). The data are first demeaned and bandpass filtered with a Butterworth filter of order

two. Each value is then calculated on a 30-seconds window (100 x 30 samples). Finally, the daily 10th and 25th percentiles and the median are typically computed to partially remove undesired transients and earthquakes which are considered a disturbance of the tremor data Di Grazia et al. (2006).

The RSAM, RSEM, SSAM, and SSEM algorithms have been implemented in an ObsPy submodule. The final code takes advantage of the resampling abilities of Pandas (Python Data Analysis Library), resulting in a total of less then 50 lines of code for the `ssxm()` method.

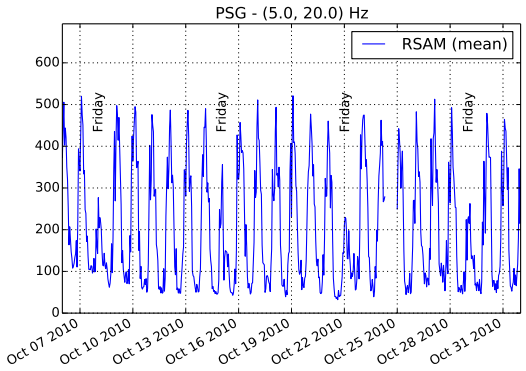

Figure 2.4.: Daily and weekly fluctuations of the seismic amplitude (RSAM) in the 5-20 Hz frequency band for station PSG during a quiet period of volcanic activity. Fridays (prayer day) are clearly visible. The amplitude is in *counts*.

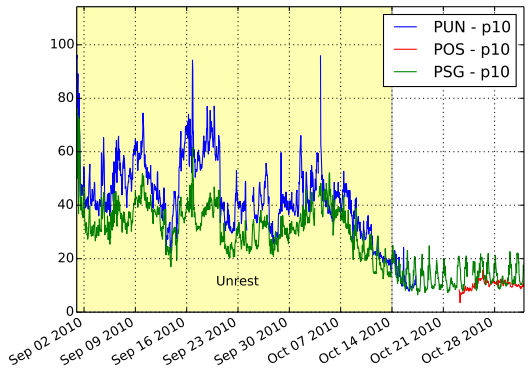

Figure 2.5.: Transition between a period of unrest to a more quiet period of volcanic activity for stations around the crater (POS and PUN) and close to the parking/base camp for climbing the volcano (PSG). The anthropic influence is still visible on PSG even when computing the 10th percentile of the amplitude of the records.

Example Results

The interest of filtering the signal in different frequency bands can be illustrated with the Kawah Ijen volcano (East Java, Indonesia) seismic data. The seismic time series are generally polluted by anthropic activities during the day, especially in high frequency bands ($>$ 10 Hz) and particularly for the stations located closer to the source of 'cultural' noise, in this case the parking and base camp for climbing the volcano. One may even observe weakened working activity during the prayer day (i.e. Friday, see figure 2.4). To minimise the effects of transients, one can use the 10th percentile (`p10`) of the amplitude in a specific frequency band (Figure 2.5). Except during volcanic unrest (Figure 2.5), the `p10` data from station PSG is still highly affected by this anthropic noise, compared to POS and PUN which are located around the crater, thus further away from the human activity.

Other studies investigated some particular processes by filtering the seismic signal. For example, SSEM computations might provide a measure of the rate of strain released, if persistent fracturing at any scale produces a sustained seismic signal De la Cruz-Reyna et al. (2010). One can also assess the stationarity of the microseism noise sources (0.1-1 Hz) to evaluate the results from the velocity variations using ambient seismic noise cross correlation techniques (Lecocq et al., 2014).

2.5 Conclusion

Legacy codes and data formats are widely used in a number of sciences and will continue to play an important role in the foreseeable future. The approaches we laid out in this article help in integrating them in modern computing environments. Most of the illustrated design and implementation choices are transferable to other fields in computational science. The extraction of common features from a heterogeneous set of file formats enables the construction of data source independent workflows which especially benefits data heavy domains. This, together with a library of functionality in a domain specific vocabulary and the inclusion of some critical legacy codes yields an attractive package even for users not familiar with Python.

ObsPy enjoys a large rate of adaption within the seismological community. We believe this is in large parts due to it solving an actual problem: the various different file formats previously requiring a plethora of file format converters or different signal processing tools. Once people start to use it they quickly discover the flexibility and power of Python. In contrast to for example MATLAB, using ObsPy also grants the advantages of a full blown programming language. A further competitive edge are the many third-party modules in Python which are not directly associated with signal processing enabling the use of for example databases, web services, machine learning libraries, and of course the recent developments in handling big data sets which will become more and more important in seismology and other fields. On top of all that the complete stack is free, open-source, and runs on virtually every platform of relevance.

Studies that successfully utilized ObsPy include event relocations (Megies and Wassermann, 2014), rotational (Hadziioannou et al., 2012) and time-dependent (Richter et al., 2014) seismology, big data processing (Atkinson et al., 2013), and synthetic studies about full-waveform inversions (Schiemenz and Igel, 2013) and attenuation kernels (Fichtner and van Driel, 2014) to name a few examples. The ObsPy website, accessible under `http://www.obspy.org` contains a detailed documentation, an extensive tutorial, and access to a mailing list in order to ease the transition to ObsPy and build a community around it. The modularity described in this manuscript and the test-driven development facilitate the addition of new functionality and as a result ObsPy steadily increases the number of external contributions with the goal of becoming a code maintained by people throughout the community.

Acknowledgements

We want the thank the steadily growing ObsPy community for the continuous support and constant encouragement and exchange. By now too many people contributed code and ideas to ObsPy to list them all but we are grateful for each and every one of them. We also thank the developers of Python, NumPy and consorts, and the authors of a number of libraries integrated into ObsPy. Two anonymous reviewers helped to improve the final manuscript with thoughtful reviews.

3
Managing Data with LASIF

This chapter switches focus from data processing to data management and handling in the context of full seismic waveform inversions with LASIF, the **La**rge-scale **S**eismic **I**nversion **F**ramework. It is a software package built upon ObsPy's foundation introduced in the previous chapter. Full seismic waveform inversion using adjoint methods evolved into a well established tool in part of the community and has seen many applications. While the procedures employed are well understood to a certain extent, large scale applications to real world problems are often hindered by practical issues.

The inversions use an iterative approach and thus by their very nature encompass many repetitive, arduous, and error-prone tasks. Amongst these are data acquisition and management, quality checks, preprocessing, selecting time windows suitable for misfit calculations, the derivation of adjoint sources, model updates, and interfacing with numerical wave propagation codes.

We developed a workflow framework designed to tackle these problems. One major focus of the package is to transparently keep track of all applied operations resulting in reproducible and, importantly, more trustworthy Earth models. The use of a unified framework also enables an efficient collaboration on and exchange of tomographic images. Most of this chapter has been published in:

Krischer, L., Fichtner, A., Zukauskaite, S., & Igel, H. (2015).
Large-Scale Seismic Inversion Framework.
Seismological Research Letters, 86(4), 1198–1207.
https://doi.org/10.1785/0220140248

3.1 Introduction

Since its development and first applications in the late 1970's (e.g. Aki and Lee, 1976; Aki et al., 1977; Dziewoński et al., 1977), seismic tomography has developed into a powerful tool to investigate the internal structure of the Earth across scales. Tomographic Earth models have become increasingly detailed thanks to the continuous densification of the global station network (e.g. Roult et al., 2010; Gee and Leith, 2011), the installation of dedicated arrays (e.g. SKIPPY (van der Hilst et al., 1994), USArray (www.usarray.org), IberArray (Díaz et al., 2009)), and the deployment of ocean-bottom seismometers (e.g. Shiobara et al., 2009; Obayashi et al., 2013). Furthermore, methodological developments have sharpened our picture of the Earth. Depending on the nature of the data, the scientific question and the available resources, seismic tomographers can choose from a rich variety of techniques, including ray tomography (e.g. Kissling, 1988; Spakman, 1991; Grand et al., 1997; Rawlinson and Sambridge, 2003), various finite-frequency methods (e.g. Yomogida, 1992; Dahlen et al., 2000; Friederich, 2003; Yoshizawa and Kennett, 2004, 2005), or full waveform inversion based on numerical solutions of the wave equation (e.g. Tarantola, 1988; Chen et al., 2007; Fichtner et al., 2009; Zhu et al., 2012; Fichtner et al., 2013; Afanasiev et al., 2014). Improvements of data coverage and inversion technology give rise to new challenges that need to be addressed in order to ensure continued progress:

(i) Exponentially growing amounts of data and metadata must be retrieved, organised, quality-controlled, and maintained up-to-date.

(ii) Data is available in many different, often purpose-tailored formats and with variable pieces of information, which makes the handling of large datasets unnecessarily cumbersome.

(iii) The growing complexity of increasingly sophisticated tools reduces our ability to independently assess the results of tomographic inversions and to collaborate across different research groups. The flood of provenance information needed to enable reproduction of scientific results is increasingly difficult to organise.

(iv) The processing of large waveform datasets, and the measurement of differences between observed and synthetic seismograms becomes too computationally demanding to be performed on a single compute core. Modern high-performance computing resources should thus be harnessed for both processing and measurements in order to avoid bottlenecks in the seismic inversion workflow and to ensure scalability.

(v) The growing complexity of hardware architectures and software developments makes it impossible for single institutions or individual researchers to maintain stable and efficient solutions for computational tasks such as seismic waveform inversion.

The development of stable community solutions plays an increasingly important role. Eventually such solutions may be merged with evolving science gateways (e.g. the EU funded VERCE project for seismology) that provide high level access to sophisticated IT applications to the scientific community. The goal of the **LA**rge-scale **S**eismic **I**nversion **F**ramework (LASIF) is to provide solutions to the above-mentioned problems, thereby reducing the time to research.

LASIF provides a flexible structure linking the different components of a tomographic inversion, including the download and processing of data, the computation of synthetics, window selection and measurements, as well as visualisation and data exploration. As such, it offers functionality for the retrieval, organisation, parallel processing, and visualisation of seismic waveform data and metadata in a variety of different formats. Furthermore, LASIF provides tools for automatic and manual window selection, the parallel measurement of differences between observed and synthetic seismograms, and for the computation of adjoint sources needed in the calculation of Fréchet kernels based on adjoint techniques (e.g. Tarantola, 1988; Tromp et al., 2005; Fichtner et al., 2006). The strict documentation of all operations performed, increases reproducibility. Through its clearly defined structure, LASIF facilitates collaborative projects. Various visualisation tools allow the user to explore data and to monitor the progress of iterative inversions. LASIF is written in Python and JavaScript, under the GPLv3 open source license and freely available online (http://www.lasif.net). The code features numerous internal testing routines that reduce the probability of programming errors, and an extensive documentation and tutorial are available online. Many routines are based on *NumPy*, *SciPy* (Jones et al., 2001), and *ObsPy* (Beyreuther et al., 2010; Krischer et al., 2015b) for the seismic analysis part.

The remainder of this chapter is organised as follows: Following a summary of LASIF's design philosophy and general structure, we describe procedures for the download of event-, waveform and station metadata. This is followed by two paragraphs on waveform processing and the link of LASIF to forward problem solvers that provide synthetic waveforms. Subsequently, we provide details on the automized selection of measurement windows, the computation of various misfit measures and corresponding adjoint sources, and on the actual inversion procedure. To demonstrate LASIF's ability to solve real-data problems, we show results of an ongoing full waveform tomography for the Japanese Islands region.

```
ProjectDirectory/
├── DATA/ ..................................................... Observed data. Raw and processed.
│   ├── event_a/
│   │   ├── raw/
│   │   └── processed_…/
│   └── event_b/
├── EVENTS/ ................................................... Event definitions as QuakeML files.
│   ├── event_a.xml
│   └── event_b.xml
├── ITERATIONS/ ............................................... Definition of the iterations.
│   ├── ITERATION_1.xml
│   └── ITERATION_2.xml
├── LOGS/
├── MODELS/
├── STATIONS/ ................................... Station information in all common formats.
├── SYNTHETICS/ ..................................................... Synthetic data.
│   ├── event_a/
│   │   ├── ITERATION_1/
│   │   └── ITERATION_2/
│   └── event_b/
├── …/
└── config.xml ..................................................... Project configuration.
```

Figure 3.1.: The directory structure of LASIF. This example misses some folders for the sake of brevity. The stateful nature of LASIF means that, as soon as some data is copied or created under it, LASIF is aware of it.

3.2 Philosophy and Structure

LASIF represents the state of a tomographic inversion in a fixed and intuitively designed directory structure on disc, summarised in figure 3.1. Tools for the modification, interpretation, bookkeeping, and visualization of the inversion infer all necessary information from the data, and modifying the data in turn modifies the state of the inversion. A number of unobtrusive caches, storing basic information about the data contained in LASIF, are employed to keep LASIF fast and responsive without getting in the users' way. These basic design principles make LASIF a data-driven framework, and they result in a number of advantages compared to approaches relying on databases or bookkeeping files:

(i) Simple installation and maintenance as no database needs to be set up and kept running, which is especially important on high-performance platforms.

(ii) Increased shareability and potential for collaboration as the fixed directory structure enables others to understand what has been done and what the next steps are.

(iii) Straightforward integration with other tools, and

(iv) simple backups, which - coupled with continuous snapshots of the file system on modern platforms - also enables recovery from and rolling back of errors.

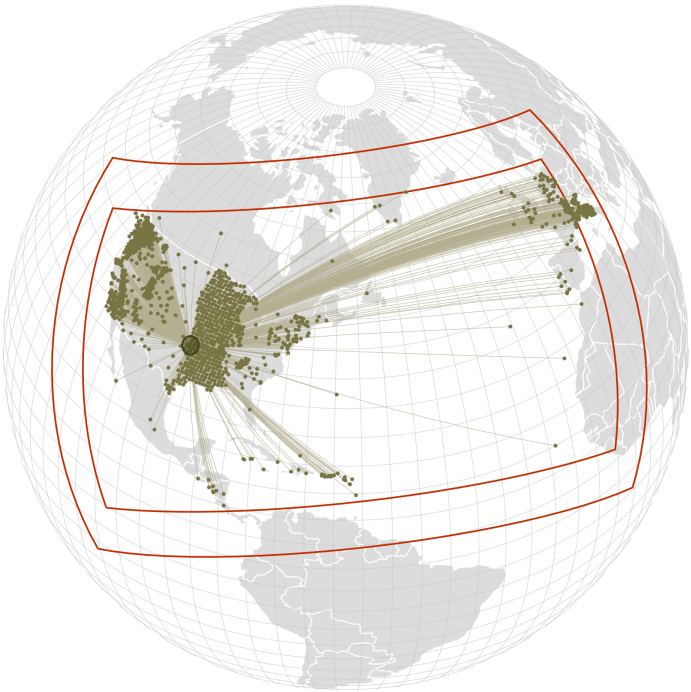

Figure 3.2.: A screenshot of LASIF's web-interface which can be launched with the `lasif serve` command at any point. The example shows the interactive map currently set to display the raypaths and recording stations for a single event. The main purpose of the web-interface is to interactively explore the data set and state of the inversion.

The internal structure of LASIF is strictly modular, with individual components being responsible for comparatively simple tasks, such as the retrieval of station and event information,

the processing of a waveform, or the calculation of a misfit. The modularity of LASIF facilitates code maintenance and the addition of new features. Modules interact with the help of three different user interfaces in order to perform more sophisticated operations:

LASIF's web interface - a screenshot of which is shown in figure 3.2 - allows the user to visually explore event and waveform data, and to monitor the evolution of synthetic waveforms in the course of an iterative inversion. The command line interface is used to steer the tomographic inversion. Executing, for instance, the UNIX shell command

```
$ lasif init_project Example
```

creates a new LASIF project entitled *Example* by setting up the directory structure from figure 3.1, as well as initial configuration files. Furthermore, the command line interface can be used to retrieve waveform- and metadata from online data centers, to preprocess data, and to automatically select measurement windows. Additional examples involving the command line interface are provided in the following paragraphs. Finally, a measurement interface can be used to select windows manually and to inspect observed and synthetic waveforms.

3.3 Event-, Waveform-, and Metadata

LASIF offers various tools for the retrieval of event-, waveform-, and metadata from online data centers. Executing, for instance, the built-in command

```
$ lasif add_gcmt_events --min_year 2005 10 5 7 250
```

will query the Global Centroid Moment Tensor project catalog (Ekström et al., 2012) in order to add up to 10 earthquakes, from 2005 or later, with magnitudes between 5 and 7, and a minimum inter-event distance of 250 km to the current project. The event distribution is optimal in the sense that it approximates a Poisson disk distribution. This is intended to generate a set of events with good data coverage and few redundancies. Each new event is chosen from all available events, by having the largest possible minimum distance to the next closest earthquake already part of the project, while still satisfying the geographic, time, and magnitude constraints. An example of automatically selected events is presented in figure 3.3.

Alternatively, individual events can be added to the project via the IRIS SPUD service (www.iris.edu/spud/momenttensor). The command

```
$ lasif add_spud_event http://www.iris.edu/spud/momenttensor/id
```

adds the event with ID `id` to the `EVENTS/` folder. All event information is written in the form of QuakeML files.

Following the retrieval of event information, waveform data can be obtained by invoking LASIF's `download_data` command. Assuming the user has defined a QuakeML file `GCMT_event_ROMANIA.xml` describing an event, then the command

```
$ lasif download_data GCMT_event_ROMANIA
```

queries a collection of FDSN web service providers and automatically downloads all waveform and station data it can find for the time frame of that event. In addition to LASIF, any other tool may be used by simply copying data into the correct folders, in this case `DATA/` and `STATION/`, respectively.

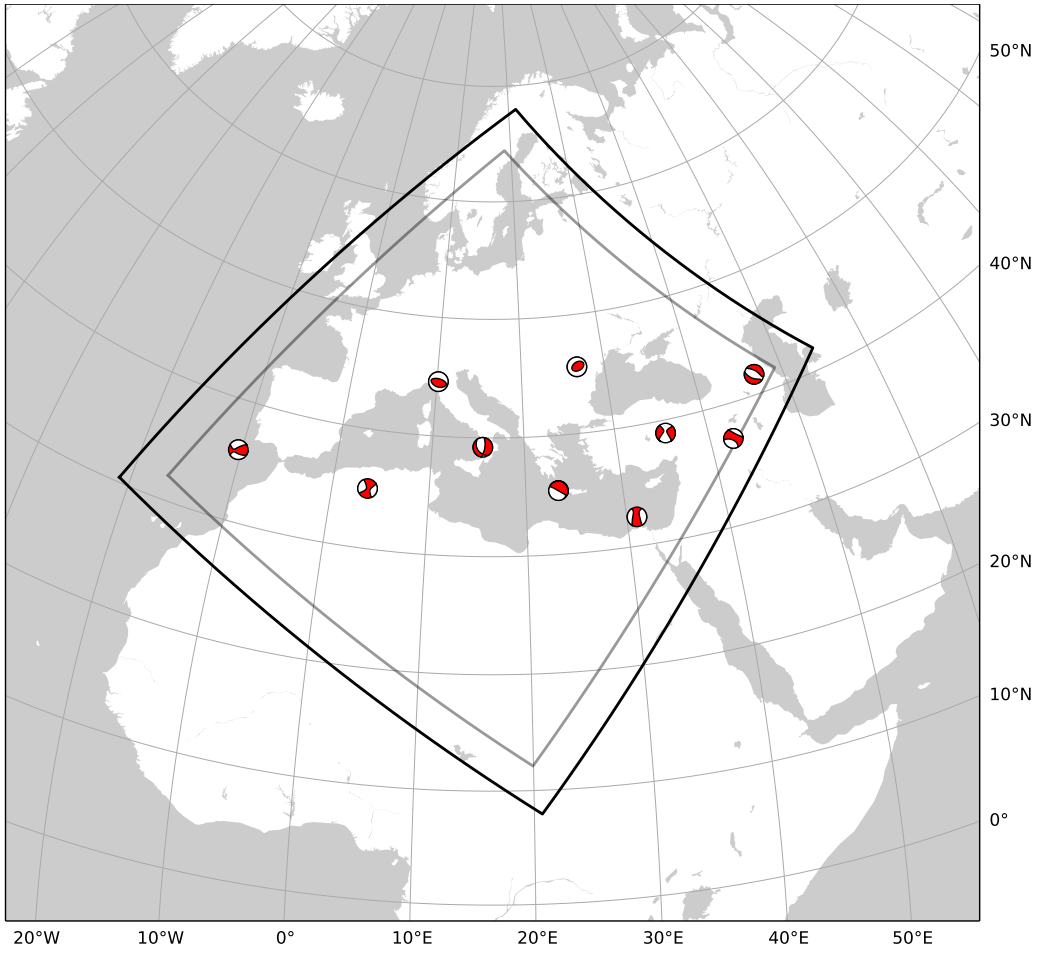

Figure 3.3.: A small set of automatically selected events. The map shows the unedited output of `lasif plot_events` which is one of several visualization commands available in LASIF. The black lines mark the boundaries of the simulation domain, the gray inner lines an optional buffer zone used to safeguard against boundary effects from numerical waveform solvers.

To honor the real world situation of multiple data providers with different standards, LASIF has been designed to be as format-agnostic as possible. While we recommend to use MiniSEED for waveform, and StationXML for station data, LASIF can also deal with SAC, GSE2, dataless SEED, RESP, and a variety of other file formats and any combination of them. This is achieved by utilizing ObsPy (Beyreuther et al., 2010; Megies et al., 2011) wherever possible. As a fallback for some combinations of waveform and station data that do not contain station coordinates, LASIF can query web services to complement the dataset with the missing information.

3.4 Data Processing

Data processing in LASIF is intended to correct and filter waveform data, and to ensure the compatibility of observed and synthetic waveforms. Taking information on the time stepping and frequency band of the forward problem's solution, the command

```
$ mpirun -n 16 lasif preprocess_data 1
```

processes all data used in iteration 1 on 16 CPUs. It can also be invoked without the Message Passing Interface (MPI) resulting in execution on only one core. The processing of observed waveforms includes the following operations:

(i) removal of the mean and linear trends,

(ii) tapering,

(iii) bandpass filtering to the frequency band used in the computation of synthetic seismograms,

(iv) removal of the instrument response,

(v) downsampling or interpolation to a sampling interval that equals the time step of the forward problem solution.

LASIF's nature enables it to make good choices for many of the parameters required for these operations. Further required information is stored in iteration XML files which are explained in the later inversion section. In order to minimise the time required for these tasks, the processing in LASIF is fully parallelised, using MPI. This parallelism allows users to process data on a large number of compute cores.

The data processing is fully configurable on a per-project and iteration basis. Furthermore LASIF can optionally process synthetic data which might be necessary depending on the specifics of the chosen inversion workflow. This processing will be applied on-the-fly anytime synthetics are required for an operation.

3.5 Synthetic Data

LASIF provides functionality to generate input files for seismic wave propagation solvers. Taking the previously compiled information about events and stations, LASIF can currently produce input files for the global spectral-element solver SPECFEM3D GLOBE (e.g. Komatitsch and Tromp, 2002b,a; Peter et al., 2011), and the regional-scale spectral-element solver SES3D (Fichtner and Igel, 2008; Fichtner et al., 2009). Thanks to the modular structure of LASIF, input file generators for other wave equation solvers can be added easily. LASIF's responsibility stops here and the users are expected to copy the input files to an available high performance computer, run the simulations, and move the resulting synthetics to the project directory managed by LASIF.

3.6 Window Selection

The selection of time windows for the comparison of observed and synthetic data is a critical aspect of seismic tomography. It strongly affects resolution, convergence, and the impact of noise on the final Earth model. In addition to the manual window selection in the measurement interface, LASIF offers an automatic window selection. Similar to FLEXWIN, developed by Maggi et al. (2009), LASIF's window selection algorithm has been originally developed for full waveform inversion applications where complete seismograms can in principle be assimilated into the inversion. However, the algorithm can be tuned in order to select, for instance, specific body or surface wave phases. It has been tested and successfully applied in inversions ranging from regional and continental scales (Fichtner et al., 2013) to the full globe.

Global rejection parameters	
`min_cc`	Minimum normalised correlation coefficient between observed and synthetic traces.
`max_noise`	Maximum relative noise level of the data trace.
Window acceptance/rejection parameters	
`min_velocity`	Minimum apparent velocity. Later arrivals are rejected.
`threshold_shift`	Maximum cross-correlation time shift within a sliding window.
`threshold_corr`	Minimum normalised correlation coefficient within a sliding window.
`min_length_period`	Minimum length of a time window relative to the minimum period.
`min_peaks_troughs`	Minimum number of extrema in an individual window.
`max_energy_ratio`	Maximum energy ratio between observed and synthetic data within a window.
`max_noise_window`	Maximum relative noise level for individual windows.

Table 3.1.: Parameters for the window selection algorithm. Correlation coefficients are normalised to range between -1.0 and 1.0, time durations are expressed as fractions of the minimum period of the input data.

The window selection operates on pairs of observed and synthetic waveforms, assuming both have been appropriately processed. In addition to the waveforms, the algorithm takes the following inputs: locations of source and receiver, the minimum and maximum period, and a set of adjustable parameters summarised in table 3.1. The algorithm proceeds in four steps that are detailed in the paragraphs below:

(i) Determination of window bounds based on travel times,

(ii) global trace rejection based on the noise level and the overall similarity between observations and synthetics,

(iii) preselection of windows based on a sliding cross-correlation, and

(iv) a number of successive elimination stages involving amplitude ratios, the minimum window length, and various other criteria.

3.6.1 Window Bounds Based on Travel Times

The first stage of the automatic window selection determines the bounds of all possible windows based on the theoretical travel times of seismic phases. The first body wave arrival computed for the 1-D Earth model AK135 (Kennett et al., 1995) marks the lower bound, and the

minimum surface wave velocity `min_velocity` (see table 3.1) the upper bound. At both ends a buffer of half the minimum period of the data is added to account for the effects of (a)causal filters.

3.6.2 Global Rejection Criteria

Prior to the detailed selection of time windows, the algorithm rejects data based on their noise level and overall similarity to the synthetics.

The relative noise level is defined as the ratio between the maximum amplitude prior to the first arrival and the maximum amplitude in the complete seismogram. Data are rejected when the relative noise level is above `max_noise`. The definition of noise is to some extent subjective. It could be improved in future versions of LASIF using, for instance, the upcoming IRIS MUSTANG service (IRIS, 2014) that is currently in the testing phase.

To ensure a basic comparability of observed and synthetic seismograms, the normalised zero-lag correlation coefficient

$$cc = \frac{\mathbf{d}^T \mathbf{s}}{\sqrt{(\mathbf{d}^T \mathbf{d})(\mathbf{s}^T \mathbf{s})}} \tag{3.1}$$

must not be lower than `min_cc`. In equation (3.1), $\mathbf{d}$ and $\mathbf{s}$ denote the arrays of observed and synthetic waveforms, respectively. A strongly negative correlation coefficient can indicate problems with the polarity, and may be used as a criterion for flipping data.

3.6.3 Sliding Cross Correlation

Provided that data pass the global rejection criteria, LASIF makes a selection of candidate windows using a sliding cross-correlation technique that is intended to avoid cycle skips. With the discrete cross-correlation between two arrays $\mathbf{f}$ and $\mathbf{g}$ defined as

$$(\mathbf{f} * \mathbf{g})[n] = \sum_m f[m] g[n - m] \tag{3.2}$$

the sliding normalised cross-correlation of observed data $\mathbf{d}_i$ and synthetic data $\mathbf{s}_i$ windowed around index i is given by

$$\mathsf{cc}_i = \frac{\mathbf{d}_i * \mathbf{s}_i}{\sqrt{(\mathbf{d}_i^T \mathbf{d}_i)(\mathbf{s}_i^T \mathbf{s}_i)}} \,. \tag{3.3}$$

The current implementation of LASIF uses a Hanning window with a length equal to twice the minimum period. Different sliding windows can be implemented with ease when needed.

At each index i, the maximum is extracted, yielding the maximum correlation at each point in time. Furthermore, the time shift is computed as the lag time where the maximum correlation occurs. A time index i is kept as a candidate index when the maximum correlation is above `threshold_corr`, and when the time shift is below `threshold_shift`.

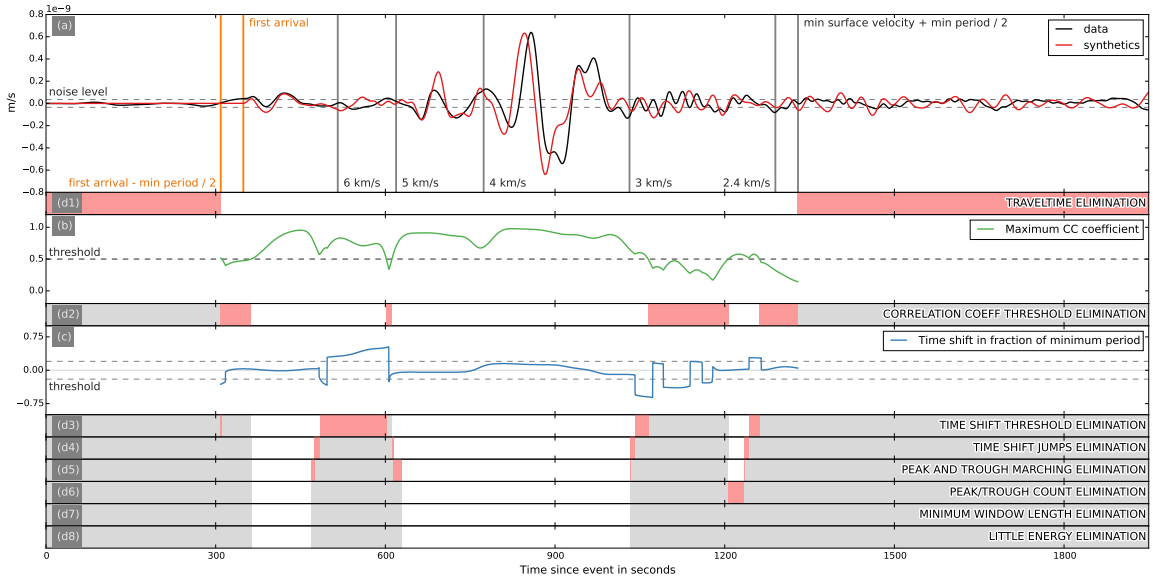

Figure 3.4.: Graphical illustration of the window selection algorithm. **(a)**: Observed and synthetic seismograms including the theoretical arrival times of the first body wave phase for model AK135 (Kennett et al., 1995) in orange. The arrival times for a range of apparent surface wave velocities are plotted in gray. The noise level estimated from the amplitudes prior to the first arrival is indicated by the gray dashed lines. **(b)** and **(c)**: Maximum windowed cross correlation coefficient and the corresponding time shift, respectively. **(d1)** - **(d8)**: Successive elimination stages of the window selection algorithm. In each stage, gray corresponds to the time intervals that have been eliminated in the previous stages. Red time intervals are eliminated in the current stage, and white corresponds to the time intervals that are still being considered. Thus, the white intervals in the bottom bar represent the final time windows.

3.6.4 Elimination Phases

The algorithm proceeds with the following elimination phases intended to exclude time intervals where observed and synthetic waveforms differ too much:

(i) A buffer around each jump in the cross-correlation time shift is marked as invalid. The occurrence of such jumps, illustrated by the blue curve in figure 3.4, is indicative of cycle skips that the algorithm attempts to avoid.

(ii) The peaks and troughs of observed and synthetic waveforms are detected by finding local extrema. Intervals where the timing of matching peaks and troughs differs by more than half the minimum period are marked as invalid. This criterion is primarily intended to detect high-frequency oscillations on top of lower-frequency data.

(iii) Windows with less than `min_peaks_troughs` local extrema are discarded.

(iv) Windows shorter than `min_length_period` are excluded.

(v) Windows where the maximum amplitude divided by the absolute noise level prior to the first arrival is smaller than `max_noise_window` are eliminated as well.

(vi) Candidate windows are kept only when the amplitudes in the ratio between observed and synthetic amplitudes is below `max_energy_ratio`.

Automatic window selection algorithms should generally not be used blindly because the - to some extent subjective - goodness of the adjustable parameters is strongly data- and application-dependent. Considering the immense amounts of waveform data that are available today, we recommend to manually tune the window selection parameters with a small subset of the data. The selection parameters can then be used to compute time windows for the remaining data. A conservative choice is generally advisable as the damage caused by inappropriate windows typically outweighs the benefit of having slightly more windows.

3.7 Misfit Measurements and Adjoint Sources

Once appropriate time windows have been selected, LASIF can compute various types of misfit measures between observations and synthetics, as well as the corresponding adjoint sources needed for the calculation of Fréchet kernels via adjoint techniques (e.g. Tarantola, 1988; Tromp et al., 2005; Fichtner et al., 2006). Executing the command

```
$ lasif finalize_adjoint_sources 1 GCMT_event_ROMANIA
```

performs this task for iteration 1 and the chosen event. For each chosen window it will calculate the misfit and derive the associated adjoint source, it will then combine all measurements for a single component, weight them, and produce the final adjoint source for that component. Weighting can be done per event, per station, and also per window. The adjoint sources will be stored in whatever format the chosen numerical waveform solver requires.

Currently implemented misfit measures include the L^2 waveform difference typically used in exploration applications (e.g. Igel et al., 1996; Pratt et al., 1998; Afanasiev et al., 2014), the cross-correlation traveltime shift used in waveform traveltime inversion (Luo and Schuster, 1991), and the time-frequency phase misfit (Fichtner et al., 2008, 2013).

The modularity of LASIF allows for the straightforward implementation of additional misfit measures, such as, for instance, multi-taper measurements (e.g. Laske and Masters, 1996; Zhou et al., 2004; Tape et al., 2010) or generalised seismological data functionals (Gee and Jordan, 1992).

3.8 Inversion

A key functionality of LASIF consists in the tracking of the inversion process through a series of iterations. When event and station information, as well as waveform data are available, a new iteration can be defined via the command line interface:

```
$ lasif create_new_iteration iteration_name passband forward_solver
```

All relevant information about an iteration is stored in a custom XML file that can be read and modified by any modern programming language. The iteration XML file contains

(i) information on the frequency passband,

(ii) a list of all stations for each event with optional weighting factors and time corrections,

(iii) the name of the forward problem solver, plus all setup parameters needed to run forward simulations.

The iteration XML files for a sequence of iterations keep a large part of the provenance information in a compact form, thereby facilitating reproducibility and collaborative inversion projects. Furthermore, the iteration XML files serve as input for the data preprocessing, the automatic window selection algorithm, the computation of misfits and adjoint sources, and numerous other functionalities of LASIF.

Progressing from the current to the next iteration, requires the generation of a successor to the current iteration XML file, as well as the translation of the current time windows to the next iteration. These tasks can be performed also via LASIF's command line interface:

```
$ lasif create_successive_iteration current_iteration_name next_iteration_name

$ lasif migrate_windows current_iteration_name next_iteration_name
```

3.9 What LASIF Does Not Do (by Design)

LASIF provides a basic functionality for the computation of iterative model updates in the form of a Python script that computes steepest-descent and conjugate-gradient updates. Given the enormous amount of different optimisation and regularisation schemes, this script is deliberately simplistic, merely outlining the general procedure involved in the computation of a model update in a gradient-based inversion scheme. Furthermore LASIF contains no means to manage and deal with the potentially massive volumes of kernels and model updates. We made these decisions for simplicity in order to keep LASIF maintainable and efficient. Thus LASIF offers no push-button solution to full waveform inversions but significantly facilitates and stabilizes them.

3.10 Application

We illustrate some of LASIF's functionality and visualisation tools with an example waveform inversion in East Asia. The study area, shown in figure 3.5, covers the Japanese islands, Taiwan, the Korean peninsula, the easternmost parts of China and Russia, Sakhalin and the majority of the Kuril Islands chain. Due to the presence of numerous plate boundaries between the Pacific, Philippine Sea, Okinawa, Sunda, Yangtze and Amur plates (Bird, 2003), the Earth's structure in the region is exceptionally complex.

Within the model domain, we selected 58 earthquakes, distributed spatially as uniformly as possible, and with magnitudes ranging between M_w 5.0 – 6.9. We obtained waveform data from all freely available seismic networks in the area, namely the Full Range Seismograph Network of Japan, the Broadband Array in Taiwan for Seismology, the South Korean National Earthquake Network and several stations from the China National Seismic Network, the New China Digital Seismograph Network, the Global Seismograph Network and the Korean Seismic Network, made available by IRIS Data Management Center. With 165 available seismic stations and 58 events, our dataset contains more than 5500 three-component waveforms. A

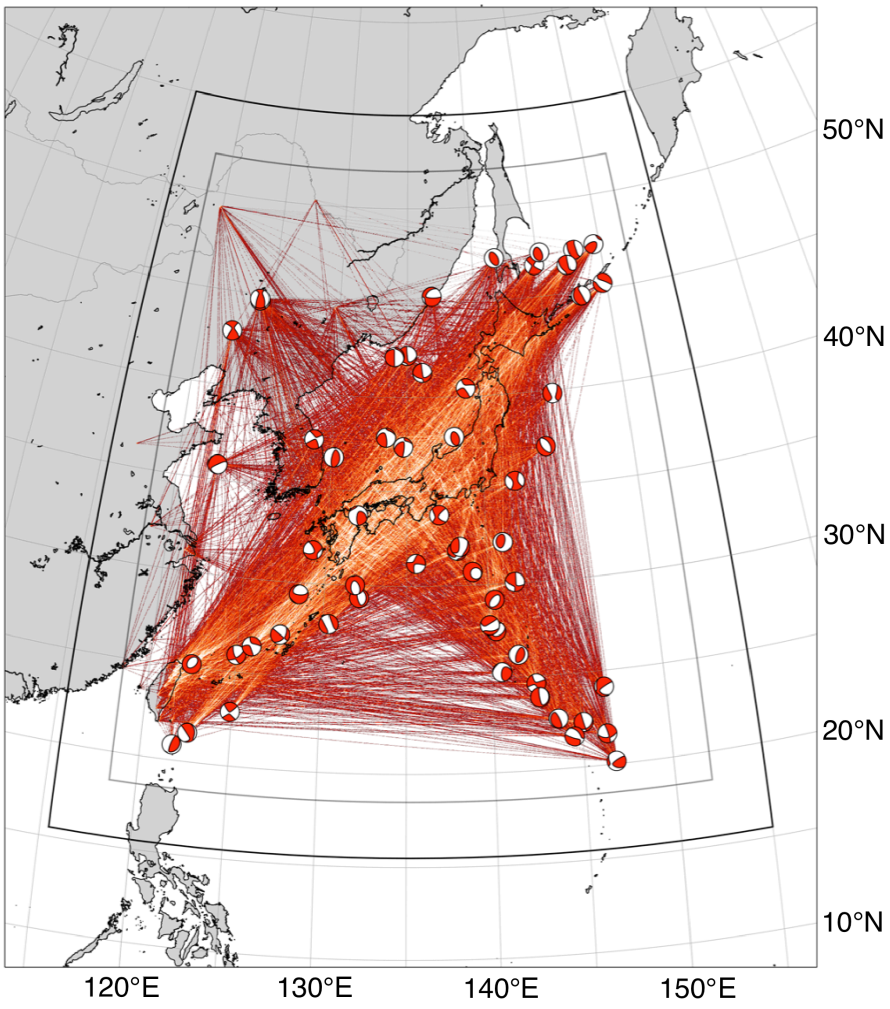

Figure 3.5.: Ray density map for the study region. Produced with the `lasif plot_raydensity` command which extracts the required information from the project file structure. Please note that it will only plot ray paths for data that is actually part of the project.

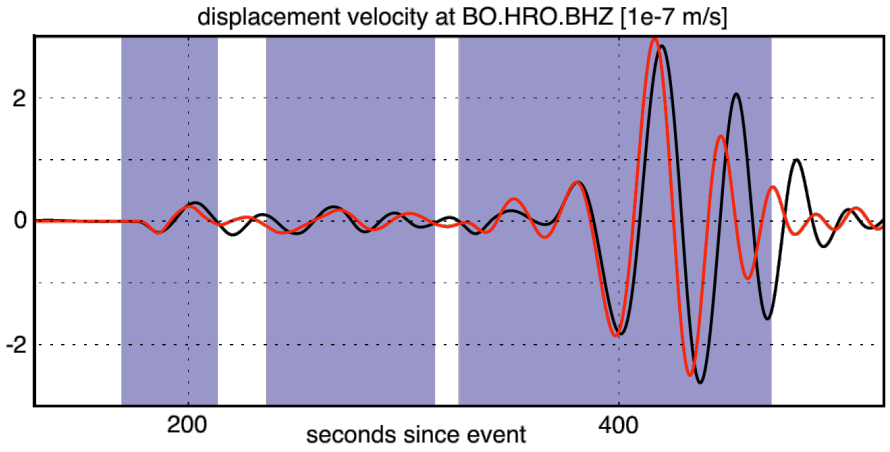

Figure 3.6.: Measurement time windows on a vertical-component velocity seismogram recorded at station BO.HRO. Windows are selected in time intervals where observed and synthetic seismograms are sufficiently close to allow for their meaningful comparison.

ray density plot that provides a first rough estimate of the achievable tomographic resolution can be produced via LASIF's command line interface (figure 3.5). For the forward simulations we use the spectral-element wave propagation code SES3D (Fichtner and Igel, 2008; Fichtner et al., 2009), run on the high-performance computers of the Swiss National Supercomputing Centre. LASIF produces all relevant SES3D input, including the geometric setup, parallelisation, visco-elastic relaxation parameters, source-time function, earthquake source parameters, and receiver positions. The automatic generation of input files for the forward solver reduces the risk of errors and facilitates reproducibility.

To ensure meaningful measurements of waveform differences, LASIF applies the same processing to observed and synthetic waveforms. Using the tunable automatic window selection described above, we determine an initial set of measurement windows that we adjust manually when needed. An example window selection as it appears in LASIF's measurement interface is shown in figure 3.6. To first constrain the long wavelength structure, we started with longer-period data filtered between 50 and 80 s. In total we selected around 4000 measurement windows where the time-frequency phase differences between observed and synthetic seismograms, as well as the corresponding adjoint sources were calculated (Fichtner et al., 2008). Taking the 3-D model of Diaz-Steptoe (2013) as initial model, we achieved a misfit reduction of 27 % after six iterations. Figure 3.7 visualises the improving match between observations and synthetics that can be monitored through LASIF's web interface.

Subsequently, we broadened the period band to 30 – 80 s, and selected around 5500 new measurement windows. Another six iterations reduced the misfits by 19 %, leading to the model displayed in figure 3.8.

Using LASIF's command line interface, the inversion procedure outlined above can be fully automized. This, however, does not mean that LASIF should be used as a black box. Human intuition remains essential for the meaningful solution of any ill-posed inverse problem, including seismic tomography. Nonetheless LASIF enabled significant speedups resulting in this inversion being carried out by a student in the course of a master's thesis.

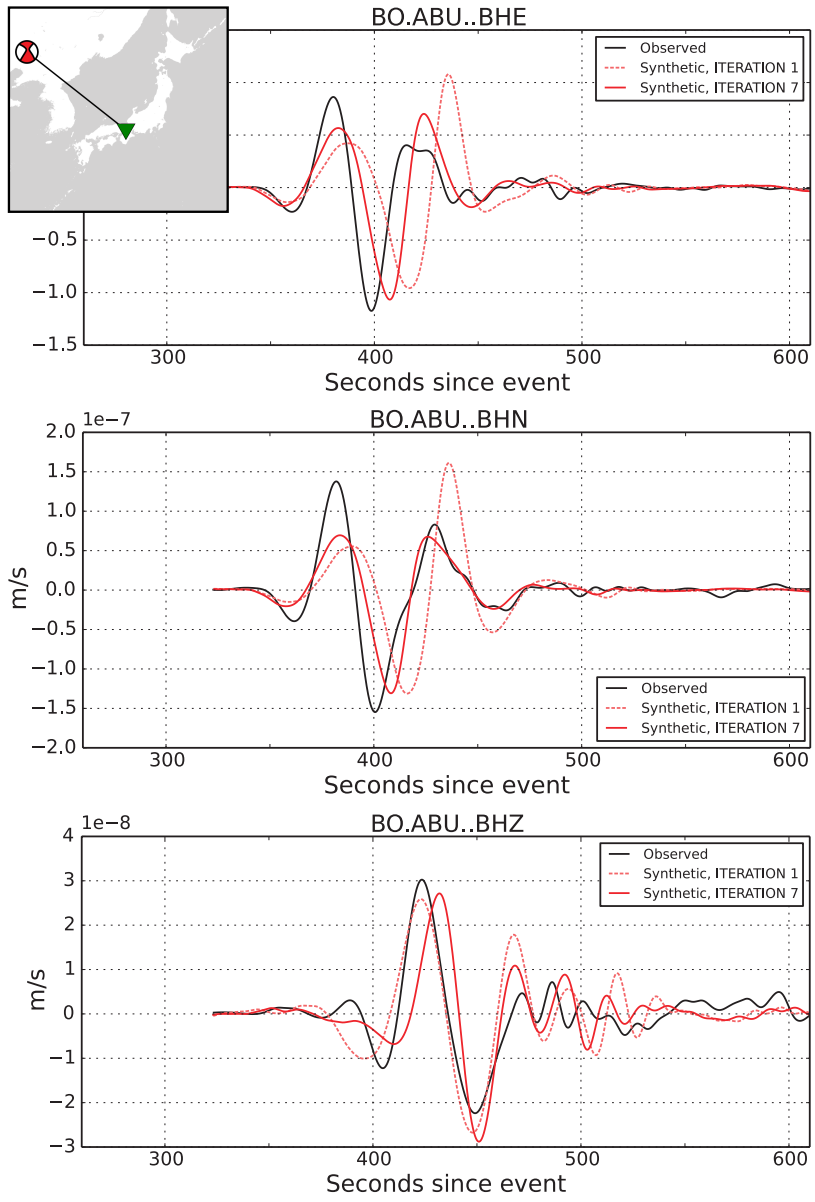

Figure 3.7.: Waveform comparison between iteration 1 (dashed light red) and iteration 7 (red) for an Mw 5.0 event in northeastern China and station BO.ABU. Observed data are plotted in black. While the waveform fit for horizontal components improves substantially, the fit in the vertical component slightly declines.

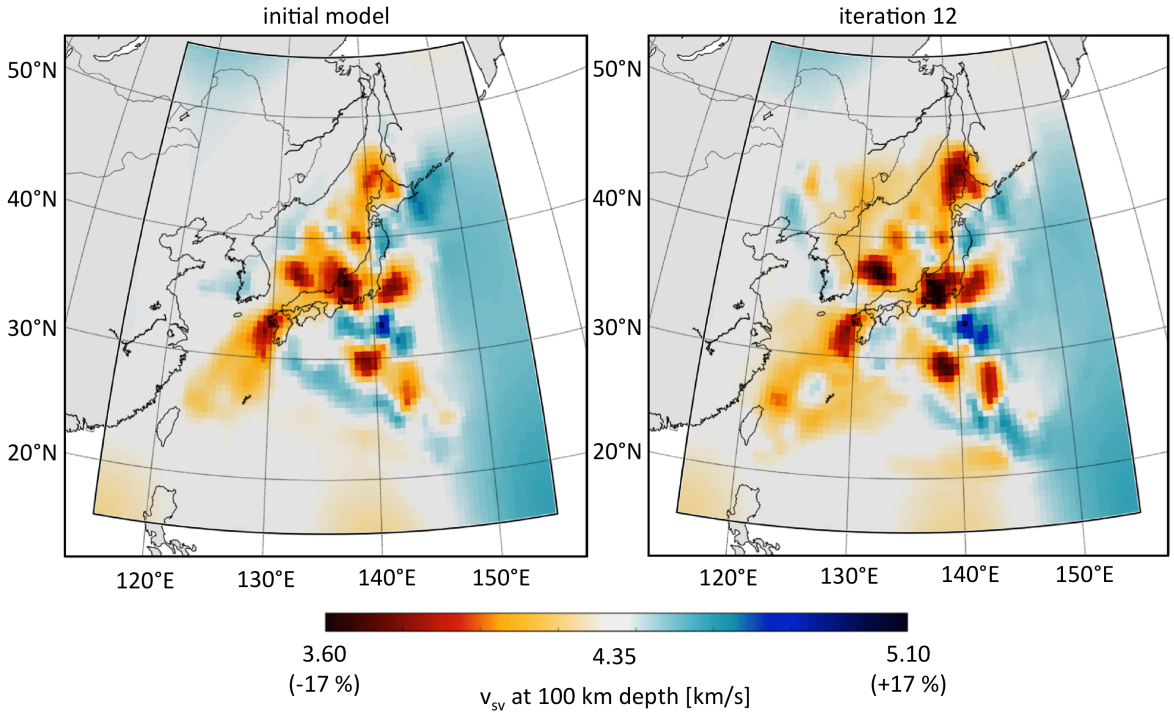

Figure 3.8.: Comparison of the SV velocity at 100 km depth in the initial model (left (Diaz-Steptoe, 2013)) and the model after 12 iterations (right).

3.11 Conclusion

We present a data management and inversion framework for potentially large-scale seismic tomography problems. LASIF is intended to increase the quality of and reduce the time to research. It does so by providing solutions to current challenges, the rapidly growing amount of seismic data, the existence of different data formats, and the decreasing reproducibility of increasingly complex inversions.

Written mostly in Python, LASIF has a modular structure that facilitates maintenance and the addition of new features. LASIF is well documented, open-source, freely available online (http://www.lasif.net), and its source code is managed via GitHub. LASIF includes

(i) tools for the download of event, waveform and station data,

(ii) a command line and a web interface to explore data and monitor the progress of an inversion,

(iii) tools for data processing,

(iv) tools for the generation of input files needed in forward problem solvers,

(v) a tunable automatic window selection algorithm,

(vi) routines for the calculation of various waveform misfit measures and corresponding adjoint sources, and

(vii) a wide range of visualisation tools.

While LASIF is a production-stage code, several future developments could still be envisioned. The incorporation of noise correlation data, for instance, currently requires a deliberate misuse of data formats that were originally designed for earthquake or active-source data. The design of a generic format for noise correlations with their complex processing history (e.g. Bensen et al., 2007), and the incorporation of this format into LASIF, has the potential to greatly improve the efficiency and reproducibility of noise tomography.

Other types of datasets with unique features, like scattered body waves used in the receiver function community, could be utilized within LASIF with only slight modifications. LASIF is independent of the numerical waveform solver, so it is straightforward to integrate for example hybrid methods (e.g. Tong et al., 2014) and define additional misfit functionals.

Furthermore, the interfacing of LASIF with a non-linear optimisation toolbox, as well as tools for the exchange of data with high-performance computers are currently being considered. The incorporation of such new features has to be weighted against the increasing complexity of the code.

Eventually it is conceivable that entire work flows such as LASIF can be offered to the community through gateways as envisaged in the VERCE project (http://www.verce.eu). In the future it is important that such software products are treated as (real) infrastructure by the communities and funding agencies with sustained support. While this might require a paradigm shift, without it we will not be able to make efficient use of the continuously expanding cyber infrastructures for our sciences.

Acknowledgements

We would like to thank the editor Zhigang Peng, as well as Qinya Liu and Carl Tape for their thoughtful and constructive reviews that helped improve the manuscript. The development of LASIF, as well as a series of pilot applications were supported by the EU-FP7 VERCE project (number 283543), the Swiss National Supercomputing Centre (CSCS) through the CHRONOS Project ch1, and by the Platform for Advanced Scientific Computing (PASC). The authors are grateful to the first users of LASIF, Michael Afanasiev, Yesim Cubuk, Erdinc Saygin, Katrin Peters, and Korbinian Sager.

4
Storing Data with ASDF

Large full seismic waveform inversions, especially those with many receivers require dealing with an enormous amount of raw, processed, and synthetic waveform data. It quickly became apparent that the pure number of files is prohibitive and a major bottleneck when performing such an inversion. Current data formats in use in seismology were just not cut out for the task at hand.

In order to create a solution not only for us but useful to the whole community we started a collaboration with Princeton University and defined a new data format for seismology, the **A**daptable **S**eismic **D**ata **F**ormat (ASDF). It is a high-performance, self-explaning data format that is capable of storing full data sets including all required meta information in a single file.

To use this, we developed two key libraries, a Python one integrating with ObsPy for the data processing and managing part and another C one to be incorporated into massively parallel waveform propagation solvers.

This chapter presents the culmination of these collaborative efforts and parts of it were published in:

Krischer, L., Smith, J., Lei, W., Lefebvre, M., Ruan, Y., Andrade, E. S. De, Podhorszki, N., Bozdağ, E., & Tromp, J. (2016).
An Adaptable Seismic Data Format.
Geophysical Journal International, 207(2), 1003–1011.
https://doi.org/10.1093/gji/ggw319

4.1 Introduction

4.1.1 Motivation

Seismology is, to a large extent, a science driven by observing, modeling, and understanding data. The process of making discoveries from data requires simple, robust, and fast processing and analysis tools, empowering seismologists to focus on actual science. Modern seismological workflows assimilate data on an unprecedented scale, and the need for efficient processing tools is pressing. In this context, the format in which data is stored and exchanged plays a central role. For example, passive seismic data are commonly stored in such a way that each time series corresponds to a single file on the file system. The amount of I/O required to process and assimilate data stored this way quickly becomes debilitating on modern HPC platforms. As another example, simulated seismograms depend on a large number of input parameters, particular versions of modeling software, and specific run-time execution commands. A modern data format should strive for complete reproducibility by keeping track of such data provenance. The majority of existing seismic data formats were created in a more primitive computing era, when no one could have foreseen the size, complexity, and challenges that seismological datasets must accommodate today (see Figure 4.1). New seismological techniques, such as interferometry and adjoint tomography, require access to very large computers, where I/O poses a major bottleneck and data mining and feature extraction are challenging.

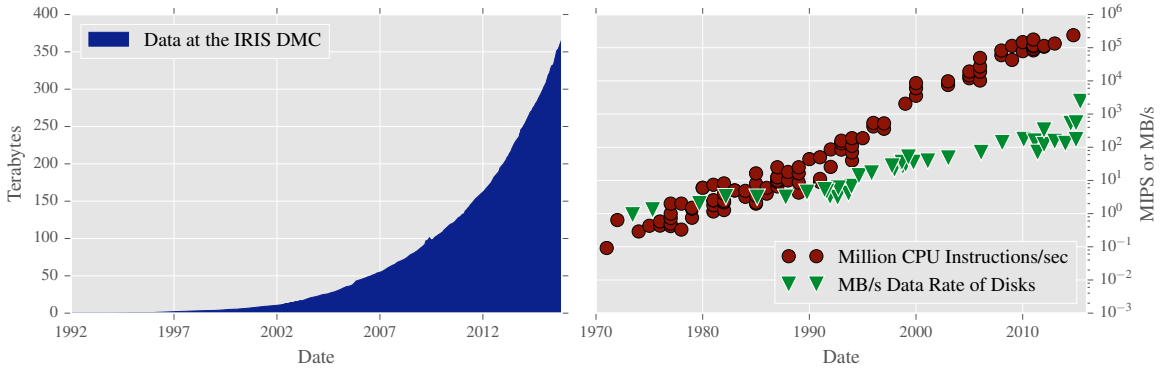

Figure 4.1.: The left panel demonstrates the nearly exponential growth of data volumes stored at the IRIS DMC (data until November 2015, used with permission from the IRIS DMC). The right-hand side illustrates the strong divergence of computing power versus I/O speed. For one, more data is becoming available and our CPUs will continue to grow faster than data can be read and written (CPU data from Wikipedia's *Instructions per second*, March 2016; Disk I/O data from Thompson and Best (2000) until 2000, more recent data collected from various data sheets and benchmarks; neither is comprehensive nor claims to be scientifically accurate but they demonstrate the trends). Both clarify and help justify the need for smarter and more efficient ways of dealing with seismological data.

In this article we introduce a new data format —the Adaptable Seismic Data Format (ASDF)— designed to meet these challenges. We are fully aware of the fact that the introduction of yet another seismic data format should ideally be avoided. However, we believe it to be justified because the current state of the art is just not good enough. We further believe that the advantages of the proposed format are significant enough to quickly outweigh the initial difficulties of switching to a new format. We identify five key issues that a new data format must resolve, namely:

- **Efficiency:** Data storage is cheap, but data operations are increasingly becoming the limiting factor in modern scientific workflows. More efficient and better performing data processing and analysis tools are badly needed.

- **Data Organization:** Different types of data (waveforms, source & receiver information, derived data products such as adjoint sources, receiver functions, and cross correlations) are needed to perform a variety of tasks. This results in ad hoc data organization and formats that are hard to maintain, integrate, reproduce, and exchange.

- **Data Exchange:** In order to exchange complex datasets, an open, well-defined, and community driven data format must be developed.

- **Reproducibility:** A critical aspect of science is the ability to reproduce results. Modern data formats should facilitate and encourage this.

- **Mining, Visualization, and Understanding of data:** As data volumes grow, more complex, new techniques to query and visualize large datasets are needed.

The ultimate goal is to empower seismologists to focus on actual science. This is the time for the community to build an organized, high-performance, and reproducible seismic data format for seismological research. In order to facilitate integration of the new format into existing scientific workflows and to demonstrate that this is not just an academic exercise, we developed a Python library hooking ASDF into the ObsPy library (Beyreuther et al., 2010), which, as a hugely beneficial side-effect, also takes care of any data format conversion issues, be it to or from ASDF. A C-based ASDF library features an API for reading and writing ASDF files and includes examples in both C and Fortran. Embedding this library in the widely used spectral-element waveform solver SPECFEM3D_GLOBE (Komatitsch and Tromp, 2002a,b) made it gain native support for ASDF-integrated workflows. To engage and educate the community, a wiki provides demonstrations of the format and includes technical and non-technical introductions for both users and developers.

4.1.2 Scope

The proposed Adaptable Seismic Data Format is designed to be an efficient, self-describing data format for storing, processing, and exchanging seismological data, including full provenance information. By extension, it is not meant to replace the time proven MiniSEED archival format used in data centers. Instead, it is a flexible data format for researchers and analysts working with the data. ASDF is applicable to a large number of areas in seismology and related sciences. Its use ranges from classical earthquake seismology to active source datasets, ambient seismic noise studies, and GPS time series. Furthermore, it is generic enough to accommodate any kind of derived or auxiliary data that might accrue in the course of a research project.

4.1.3 Benefits

A well-defined format with the previously listed attributes directly results in a number of advantages and applicable use cases. In this section we list a few of these, in no particular order.

- Seismological datasets usually contain waveform data as well as associated meta data, such as information about events and stations. All this data needs to be integrated and accessed concurrently, which requires a large amount of bookkeeping as datasets grow. Consequently, many tools are one-off scripts that cannot be reused for subsequent projects. Additionally, datasets become difficult to share with research groups that do not employ the same internal structure and data organization. Over the years, numerous groups have developed customized seismological data formats to work around these limitations. In contrast, ASDF is a well-defined format that can be used to store and exchange full seismological datasets, including all necessary meta information.

- It is oftentimes convenient to locally build up a database of preprocessed waveforms. A common example is storage of instrument corrected and bandpass filtered data. If a project continues for some years, it might ultimately no longer be known how exactly data were processed. The make up of the team may have changed, or perhaps the processing software had a bug that has been fixed in the meantime, and this may or may not have affected the data. *Provenance*, that is, the tracking and storing of the history of data, solves this particular problem, and ASDF accommodates that. Existing data formats do not (or only in a very limited manner) track the origin of data and what operations were performed on it due to limited and inflexible metadata allowances. ASDF is capable of storing the full provenance graph that resulted in a particular piece of waveform or other data.

- For the first time, ASDF accommodates proper storage and exchange of synthetic seismograms, including information about the numerical solver, earthquake parameters, the Earth model, and all other parameters influencing the final result. Waveform simulations at high frequencies and in physically plausible Earth models are extremely expensive computationally, so preserving and carefully documenting such simulations is of tremendous value.

- ASDF greatly reduces the number of files necessary for many tasks, because a single ASDF file can replace tens to hundreds of thousands of single waveform files. Beside raw performance and organizational benefits, this also facilitates workflows that run into hard file count quota limits on supercomputers. Please note that ASDF can store data from very many receivers as well as arbitrarily long time series from only a single receiver and any combination in between.

- Importantly, ASDF offers efficient parallel I/O on modern clusters with the required hardware. This facilitates fully parallel data processing workflows that actually scale.

- ASDF offers optional and automatic lossless data compression, thereby reducing file size.

- Seismograms are certainly not the only type of data used in seismology. Other data types, including various spectral estimations, cross correlations, adjoint sources, receiver functions, and so on, also benefit from organized and self-describing storage.

ASDF is intended as a container for all the various kinds of data materializing in seismological research, including all required meta information. Additionally, each piece of data should be able to describe itself and what led to it. Having an organized and standard data

container will, in the long run, increase the speed and accuracy of seismic research, and provides a medium for effectively communicating research results. The remainder of this article is structured as follows. We first provide an overview of the layout of the format and justify some choices that needed to be made. We then compare the ASDF format to existing data formats in use in seismology, thereby further justifying its development. Finally, we showcase a number of existing implementations, detail several use cases for the ASDF format, and discuss future possibilities. The article is intentionally light on technical details to focus on a high-level view. A technical definition of the ASDF format can be found online.

4.2 Overview of the Format

ASDF, at its most basic level, organizes its data in a hierarchical structure inside a container — in a simplified manner a container can be pictured as a file system within a file. The contents are roughly arranged in four sections, as follows.

1. Details about seismic events of any kind (earthquakes, mine blasts, rock falls, etc.) are stored in a QuakeML document.

2. Seismic waveforms are sorted in one group per seismic station together with meta information in the form of a StationXML document.

3. Arbitrary data that cannot be understood as a seismic waveform is stored in the auxiliary data section.

4. Data history (provenance) is kept as a number of SEIS-PROV documents (an extension to W3C PROV).

Existing and established data formats and conventions are utilized wherever possible. This keeps large parts of ASDF conceptually simple, and delegates pieces of the development burden to existing efforts. The ASDF structure is summarized in figure 4.2 and is discussed in more detail in the following paragraphs.

4.2.1 Container

Large parts of the ASDF definition are independent of the employed container format. An advantage of this approach is a certain resilience to technological changes as major pieces of ASDF can in theory be adapted to other container formats. Nonetheless, the container format has to be fixed to not severely affect interoperability and ease of data exchange. We evaluated a number of possibilities and chose HDF5 (Hierarchical Data Format version 5) (The HDF Group, 1997–2015). It is used in a wide variety of scientific projects and has a healthy and active ecosystem of libraries and tools. NetCDF 4 (Rew and Davis, 1990) is implemented on top of HDF5 and ASDF does not gain from the additional functionality. While not being as fast as ADIOS (Liu et al., 2014) for the most extreme use cases, HDF5 also fulfills our hard requirement of being capable of efficient parallel I/O with MPI (message passing interface) (MPI Forum, 2009). It can be argued that seismology does not have to deal with the same amount of data as, for example, particle physics or biology, where single datasets can easily attain volumes of multiple petabytes (Bird et al., 2014; Stephens et al., 2015). At the time of

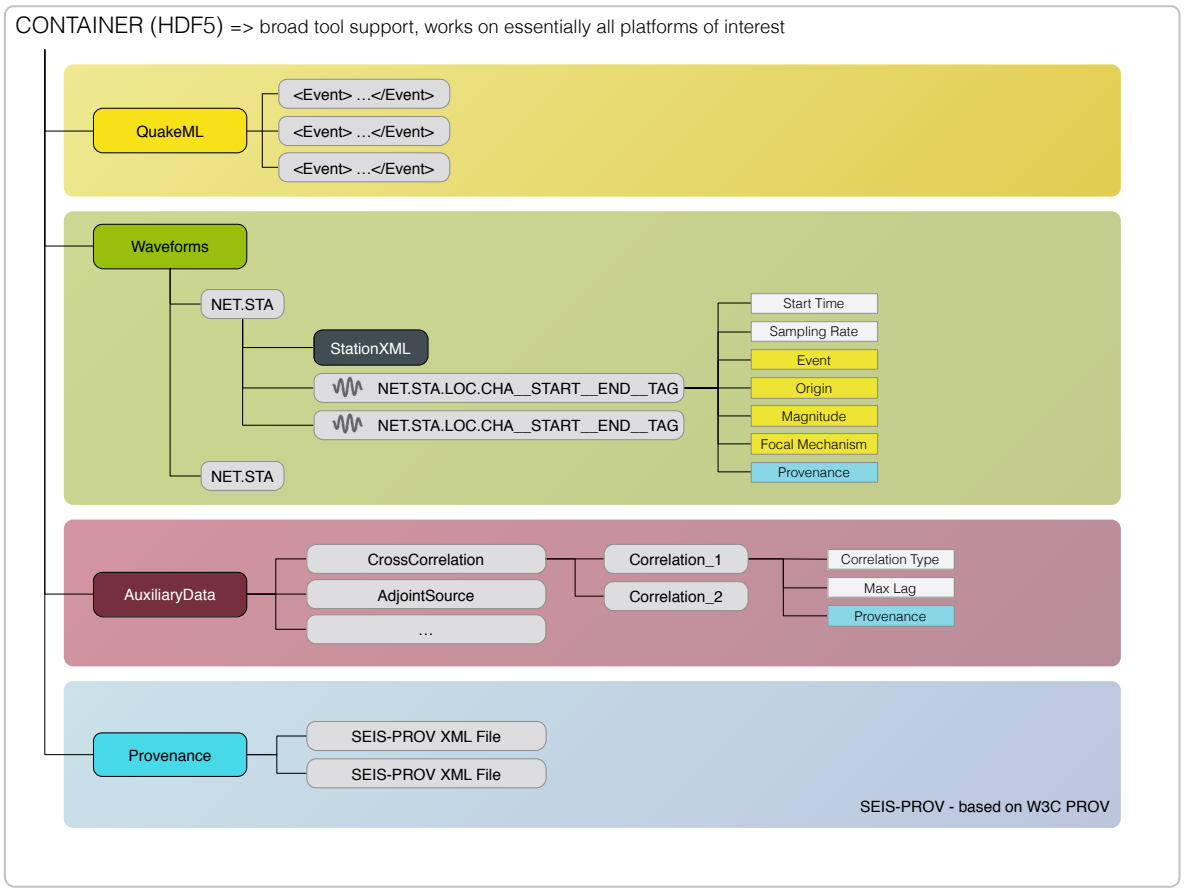

Figure 4.2.: HDF5 container — it has four distinct parts: (1, yellow) Information about an arbitrary number of earthquakes (or other seismic events) is stored in a single QuakeML document, the most complete earthquake description format currently available. (2, green) Seismic waveforms are stored per station together with the necessary meta information in the form of an FDSN StationXML document. (3, red) Anything that cannot be regarded as a seismic waveform is hierarchically stored in the auxiliary data section. (4, blue) Provenance information is stored as a number of SEIS-PROV documents, an extension to W3C PROV. Background colors in the attributes (rectangular boxes) denote relations to other sections in an ASDF file. Examples of this are relations of a waveform to a certain event or a provenance record for a piece of auxiliary data.

writing, the HDF5 libraries work on more platforms and have more users as well as available tools, which we believe is well worth the minor loss in maximum potential I/O performance. Using HDF5 also grants a number of useful features (other formats also offer some or all of them): First, there is no need to worry about the endianness of data, which historically has been a big issue in seismology. Second, HDF5 has a number of built-in data compression algorithms and data corruption tests in the form of check summing.

4.2.2 Seismic Event Information

Information about all kinds of seismic events, including earthquakes, building collapses, fluid injections, and so on, are stored in a single QuakeML (Schorlemmer et al., 2004, 2011) file inside the container. QuakeML is an XML (Bray et al., 2008) representation intended for different types of seismological meta information, but is in practice mostly used to describe earthquakes.

Note that one QuakeML document can describe an arbitrary number of events in a comprehensive manner. It is the de-facto standard for defining seismic events, adopted as a standard by the International Federation of Digital Seismograph Networks (FDSN, `https://www.fdsn.org`), and widely available, because it is served by web services of data centers around the world. A crucial capability is that it can specify a number of different hypocenters and focal mechanisms for each individual event, which might be the results from different source inversion algorithms. Each of these is identified by a unique id. ASDF uses these identifiers to determine the exact moment tensor and event location that was used to simulate an event that resulted in a particular waveform.

Shortcomings of the latest QuakeML version at the time of writing include no proper possibility for storing either finite fault sources or custom source time functions. This might be alleviated in future QuakeML versions, at which point ASDF also gains that functionality. As of now, both could either be stored in custom elements in a QuakeML document in a separate namespace, or as part of the auxiliary data section of ASDF files.

The exploration community employs seismic sources that cannot be appropriately described by the QuakeML standard. Nonetheless the concept of having detailed descriptions of seismic sources naturally translates to the active source case. It is conceivable that a standard for describing these sources might appear in the future at which point it can be incorporated into ASDF. In the use cases section we show how that can be achieved.

4.2.3 Waveforms and Station Meta Information

At the heart of ASDF is the waveform data. A single file can store any number, combination, and length of waveform data. Waveforms are restricted to single and double precision floating point and signed integer data. We make use of the universally employed SEED (Incorporated Research Institutions for Seismology (IRIS), 2012) identifiers to logically group waveforms: the network code denotes the operator of a seismological network, the station code denotes a station within that network, the location code denotes a particular instrument at a station, and finally the channel code denotes the recording component. These codes, together with some temporal information, allow the unique identification of seismic instruments and are also used in the QuakeML and StationXML standards.

ASDF organizes waveforms and associated meta information at a station level granularity. Other choices would have been possible, but this provides a certain balance between the necessary nesting and the number of elements per group (like a directory in HDF5 terms). Each station can optionally contain a StationXML document made up of meta information for one or more channels of that station. StationXML is the current FDSN standard for station information and the successor of the SEED standard. Roughly speaking, it contains information about who runs a network and deployed the station, about the geographical and geological setting of the station, and the impulse response of each recording channel. This is vital information, and storing it alongside the actual waveform data eases many common undertakings. A StationXML document can contain as much or as little information as appropriate for any given task. A further benefit is that StationXML can also be used to describe non-seismological time series, such as pressure and temperature curves.

The waveform data are stored as pieces of continuous, well-behaved time series data. Each piece, in the following called a trace, consists of a start time, a sampling rate, and a data array representing regularly sampled data. The starting time of each trace is internally represented as a nanosecond precision UNIX epoch time. The use of a 64-bit integer grants a temporal range from about the year 1680 to 2260, which is sufficient for all envisioned use cases. Times are always in UTC in accordance with most other seismological data and file formats.

Every station can contain an arbitrary number of traces containing data from multiple locations and channels. Each trace is named according to the following scheme:

```
NET.STA.LOC.CHA__STARTTIME__ENDTIME__TAG
```

`NET`, `STA`, `LOC`, and `CHA` are placeholders for the network, station, location, and channel codes. `STARTTIME` and `ENDTIME` are string representations of the start and end time of the trace. The final `TAG` part serves as another hierarchical layer. The need for this layer becomes obvious, for example, when attempting to store data from two waveform simulations but with a slightly different Earth model. They need to be given different names — a randomized string would have been possible, but human readable tags seem to be a nicer alternative. Unprocessed data straight from a digitizer are, by convention, given the tag `raw_recording`; other tags will always depend on the use case. Traces may have any length without inhibiting the ability to work with them. Incidentally, HDF5 supports reading portions of an array which enables users to read only portions of very long time series within an ASDF file.

Real world data is not perfect, and seismic receivers can fail and thus produce gaps or overlaps in data. Many existing file formats have no concept of this and thus require workarounds. In ASDF a gap is represented by one trace before and another trace after the gap and two overlapping traces denote an overlap. This construct has proven itself to work very well in practice and is also employed in the MiniSEED format as well as the ObsPy library.

Last but not least, each trace potentially also carries some more meta information and relations to other places within an ASDF file. These are elaborated upon in a later section.

ASDF's construction is not a perfect fit for active source exploration data, which is mainly a consequence of the chosen nesting structure and StationXML heavily leaning towards passive source and station based seismology. Most branches of seismology, however, work with the concept of sources and receivers. Thus we encourage the exploration community to come up with a general definition of their receivers, at which point it can be integrated into ASDF with only a minor effort.

4.2.4 Auxiliary Data

Seismologists are used to working with waveform data so they oftentimes exploit the same formats for other data. Receiver functions, cross correlations, and H/V stacks are all examples of this reuse. Header fields of the format are then used to store some limited amount of meta information. This becomes problematic if that data should be archived for future generations of researchers or exchanged with the wider community. Within the ASDF format this type of data is referred to as auxiliary data, and can be anything that is not considered a seismic waveform. Conceptually, each piece of auxiliary data is stored in an arbitrarily nested path in the auxiliary data group and consists of a data array of any dimension and any necessary meta information in a key-value representation.

ASDF does not define auxiliary data in more detail on purpose. On the one hand, many areas of seismology where the concept of auxiliary data is interesting are in a heavy state of flux and are seeing a lot of active research. It is often unclear what to store and keep track of and that view constantly evolves. On the other hand, we are not experts in all areas of seismology, and it would take a long time to agree on what needs to be stored for each type of auxiliary data.

Over time, we hope that conventions for certain types of data, such as cross correlations, will become established by the wider community. Nonetheless, ASDF allows for arbitrary and descriptive meta information for any type of data to explain what the data actually is. This becomes particularly powerful when combined with the provenance information, which is described next.

4.2.5 Provenance

Reproducibility is frequently discussed and widely recognized as a critical requirement of scientific results. In practice, it is so difficult and time consuming to achieve that it is frequently just ignored. Provenance is the process of keeping track of and storing all constituents of information that were used to arrive at a certain result or a particular piece of data. This information is then used to judge the quality and trustworthiness of the results. While not being identical to reproducibility, the concept of provenance is a key ingredient towards this goal.

Each piece of waveform and auxiliary data within ASDF can optionally store provenance information in the form of a W3C PROV or SEIS-PROV document. The implications of this are that ASDF can store any piece of observed, processed, derived, or synthetic data with full provenance information. Thus, such a file can be safely archived and exchanged with others, and information that led to a certain piece of it is readily available.

W3C PROV is a data model to describe provenance, and SEIS-PROV is a domain-specific extension for using W3C PROV in the context of seismological data processing and generation. We quickly introduce SEIS-PROV as it is a critical component of ASDF; the motivation and reasoning behind it will be detailed in a separate publication. Some examples of its use are shown in the later use cases section.

Provenance can be described from different points of view. SEIS-PROV employs a process-centered provenance description that aims to capture all actions taken to arrive at a certain piece of data. That is a natural fit for seismological data processing. In a nutshell, it works by describing things or entities which (in the context of seismology) might be waveform traces or cross correlation stacks at different stages in a processing chain. These representations are then connected by so called activities that can use existing entities and create new ones. A simple

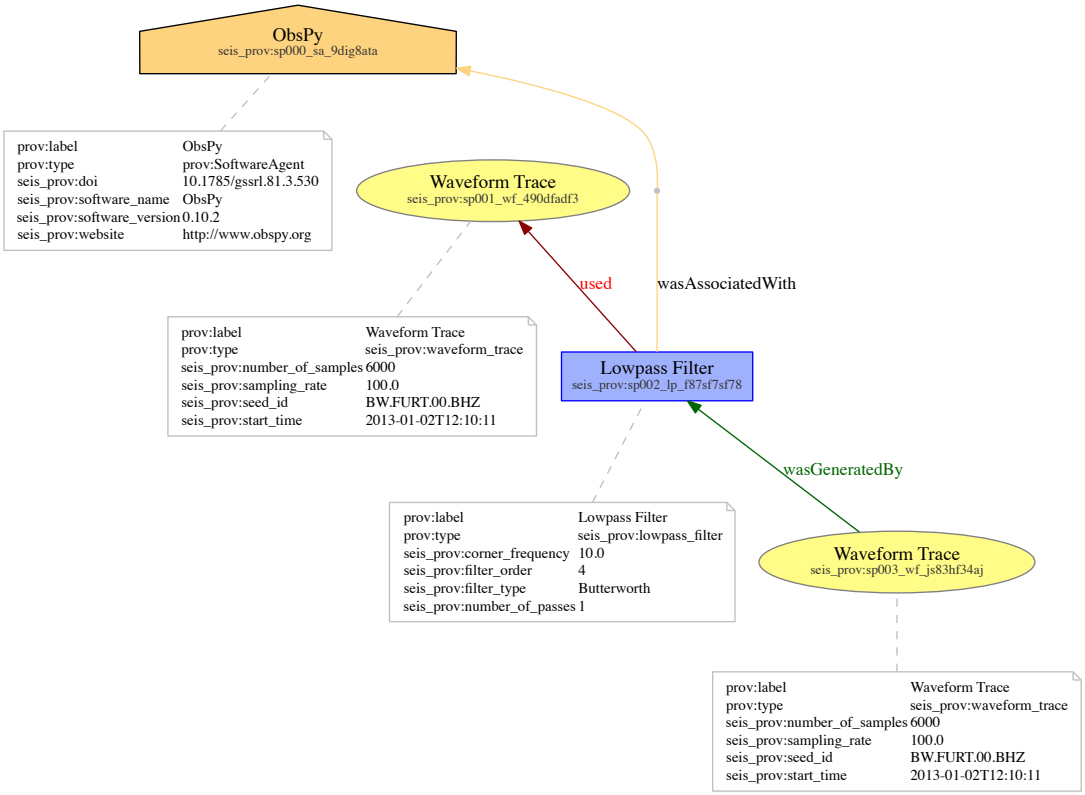

Figure 4.3.: Simple example to illustrate the key concepts of storing provenance information with SEIS-PROV and W3C PROV. It describes a single waveform trace that has been lowpass filtered to create a filtered waveform trace. The arrows in this graphical representation mostly point backwards in the process towards the origin of something. The yellow ellipses are called entities, and here they represent a waveform trace at two different points in time. The blue rectangle is an activity that can use and generate entities. It denotes a lowpass filter and uses the first waveform trace to generate a new, filtered waveform trace. The orange house shape symbolizes an agent who is responsible for something. In this case it stands for the software that performed the filtering operation. Finally, the white rectangles are attributes with more details about any node. Please note that this figure shows only one possible graphical representation of the underlying data model and more or less detailed ones can be employed as appropriate.

example of an activity is a filter in signal processing that takes an existing waveform trace and produces a new, filtered one. Additionally, all entities and actions can be assigned to agents that are responsible for it. Agents are usually persons or software programs. Figure 4.3 illustrates these concepts with a simplistic example.

The goal of the provenance descriptions in ASDF is that scientists looking at data described by it should be able to tell what steps were taken to generate that particular piece of data.

ASDF only cares about the storage of the provenance information. In practice the provenance will only be generated and used if it is captured and stored in a fully automatic fashion and is thus strongly dependent on the software used to generate and process data.

4.2.6 Data Relations

Data always needs to be regarded and interpreted in a wider context. This ranges from information about the origin of the data, which is dealt with in the previous section, to relations to other pieces of data. Classical relations in seismology are waveform data and information about the recording site and instrument, as well as the sources of the recorded wavefield.

Any time different pieces of data are required that are stored in varying places, formats, and files, the required bookkeeping to make workflows run can be substantial. ASDF greatly eases that pain by storing everything in a well-defined place within the same file. The need to find and assemble the different pieces can thus be performed by software, thereby requiring less mental work from the scientists. ASDF, as shown in the previous sections, can store waveforms, events, station meta information, provenance, and auxiliary data all in the same file. Additionally, it permits relations between these items. For example, each waveform trace can be associated with a certain event, or a certain event origin or focal mechanism. Relations for each block of data to its provenance record are also retained.

All in all this allows for fully self-explanatory, complete datasets preserving complex internal relations. This is something that is constantly required in scientific and data driven applications. Today, people usually deal with this by using project-specific directory structures that cannot be exchanged nor properly archived, and ASDF clearly improves that system on all fronts.

4.3 Comparison to Existing Formats

Having yet another format induces more complexity and, potentially, noise into the community using that type of data and the landscape of software able to deal with it. "Do we really need a new format?" is thus a natural and understandable question. This sections addresses why no single existing data format in seismology is able to satisfy our needs and thus justifies the introduction of the ASDF format.

We limit ourselves to detailing alternative waveform formats as we directly incorporate the StationXML and QuakeML formats and no true alternative to storing derived data or provenance is currently in existence. A wide variety of different seismological data formats is used by researchers world wide. We will discuss the most widely used ones, namely, (Mini)SEED, SAC, and SEG Y/PH5. Please see Bormann (2012) and Havskov (2010) for additional information and descriptions of more formats.

4.3.1 MiniSEED

The Standard for the Exchange of Earthquake Data (SEED) was developed in the late 1980s and at least the data-only part (MiniSEED) continues to be in wide use today, and will likely continue to be the dominant data archival format for the foreseeable future. The ASDF format does not attempt to replace it, as that effort would be futile. Some of MiniSEED's features, such as the ability to build up large data volumes by concatenating small and short pieces, are very well suited for their use in data archives, where data is constantly streamed in. While the full SEED format can in theory store waveforms as well as station meta information, the complexity of the format hinders that. It furthermore can only properly store raw waveform recordings and no event information. Additionally, the dataless part of SEED, e.g., the part with the station information, sees declining usage nowadays with that responsibility being taken

over by StationXML. MiniSEED, on the other hand, is more than capable of storing arbitrarily large waveform volumes, but the file then contains no index of what is in it, so one must always read the entire file to figure that out, making large data volumes fairly impractical. Additionally, the amount of meta information in MiniSEED files is strongly limited, so one always needs additional files to work with it.

For all these reasons, MiniSEED is still a pretty good data archival format for data centers, but it is not well suited for the later research and processing stages, where ASDF has significant advantages.

4.3.2 SAC

The Seismic Analysis Code (SAC, Helffrich et al. (2013)) introduced a new format named after its parent program, and is still in widespread use today. This is likely due to two reasons: the popularity of the SAC program itself and the relative simplicity of the format with a number of header fields that can be adapted to different purposes.

The SAC format is well suited for many tasks, but ASDF offers quite a number of advantages. The most obvious ones are the ability to store multiple components —including gaps and overlaps— in a single file without awkward workarounds, as well as the potential to create full datasets incorporating all necessary meta information. ASDF is, for large workflows, also more efficient, facilitated the storage of different data types — integers as well as floats — and, with the help of HDF5 offers file compression and check summing.

The combination of these factors results in ASDF being a lot more suitable and convenient for many workflows. Some, for example experiments with millions of waveform files, are almost impossible without a more advanced seismological data format. In fact, part of the motivation for developing ASDF stems from the fact that reading and writing SAC files for a large tomographic inversion practically brings a huge parallel file system to its knees.

4.3.3 SEG Y and PH5

The SEG Y Data Exchange Format (SEG Technical Standards Committee, 2002) is one of many in the family of data formats introduced and defined by the Society of Exploration Geophysicists (SEG) Technical Standards Committee. Amongst these, it is probably the most widely known and used. The more modern PH5 (IRIS/PASSCAL Data Group, 2012) format has a data model similar to SEG Y, but stores its data in an HDF5 container. This eliminates some limitations of the SEG Y format and facilitates more extensive meta information. It has been developed as an archiving format for active source seismic experiments. Typical workflows extract data from PH5 and save it as SEG Y, which is used in the further stages.

Both on- and off-shore active source data is very structured, meaning that all receivers generally have the same response and record for the same time span with the same sampling rate. Receivers are placed in lines and geo-referenced by relative coordinates. In contrast, passive source seismology is frequently very unstructured, with different receiver types scattered across a geographical region, and the meta information is fairly rich and detailed.

SEG Y and PH5 are well suited for active source experiments, but it is difficult to adapt these formats for passive source seismologists to suit their purposes. Historically, SEG Y is essentially not used in passive source seismology, and there is no reason to expect this will change with PH5. The inverse is true as well, in that passive source seismology tools are rarely used in active

studies. A consequence is that the current iteration of the ASDF format is not fully suitable for exploration studies as it relies on certain formats and conventions. In the use cases section we will show an example of how it can still be done.

The concept of seismic sources and receivers nonetheless holds true in both active and passive source seismology. We have the hope that, in the future, ASDF will be used as a standard for both. Active source seismology currently lacks community accepted standards for sources and receivers as is common in passive source seismology with formats like QuakeML and StationXML. Methods, ideas, and techniques are frequently exchanged between these communities, and we encourage the development of these missing standards. A common data format would enable greater sharing of tools, whole workflows, and most importantly human knowledge and skill, greatly benefiting both sides. The ASDF format is ready to incorporate these aforementioned definitions.

4.4 Implementations

This section discusses practical implementations of ASDF. First, we demonstrate how it can be integrated with codes that simulate seismic wave propagation based on a C API with Fortran bindings. Second, we show how it can be seamlessly integrated with Python-based analysis codes, such as the ObsPy library. Technological advances often make existing codes and tools obsolete in a matter of just a few years, and we anticipate that these implementations will continue to undergo rapid development and expansion.

4.4.1 C API with Fortran Bindings

The asdf-library is a C library with Fortran bindings that is intended to be integrated into high-performance codes. It offers support for reading and writing ASDF files in a parallel fashion by exploiting the capabilities of modern HPC and I/O systems. This application program interface (API) is the backbone of ASDF support in the widely used SPECFEM3D_GLOBE (Komatitsch and Tromp, 2002a,b) wave propagation solver. Some applications of these capabilities are illustrated in the use cases section.

4.4.2 Python Library

We also developed a Python library, called pyasdf, for working with ASDF files. It is based on the ObsPy library (Megies et al., 2011; Krischer et al., 2015b), and thus has access to a vast array of processing and I/O tools.

One of pyasdf's key features is its capability to convert ASDF to and from essentially every widely used data format in seismology, thereby greatly easing migration to the new format. Additionally, it can conveniently read existing and create new ASDF files and possesses querying capabilities, meaning one can search for certain things in an ASDF file, effectively creating a light-weight database system. pyasdf has been developed with the goal of enabling efficient processing of large datasets. Thus it contains functionality to readily process files in parallel, either with MPI or on parallel shared-memory machines. If run under MPI, it will utilize parallel I/O on machines that support it.

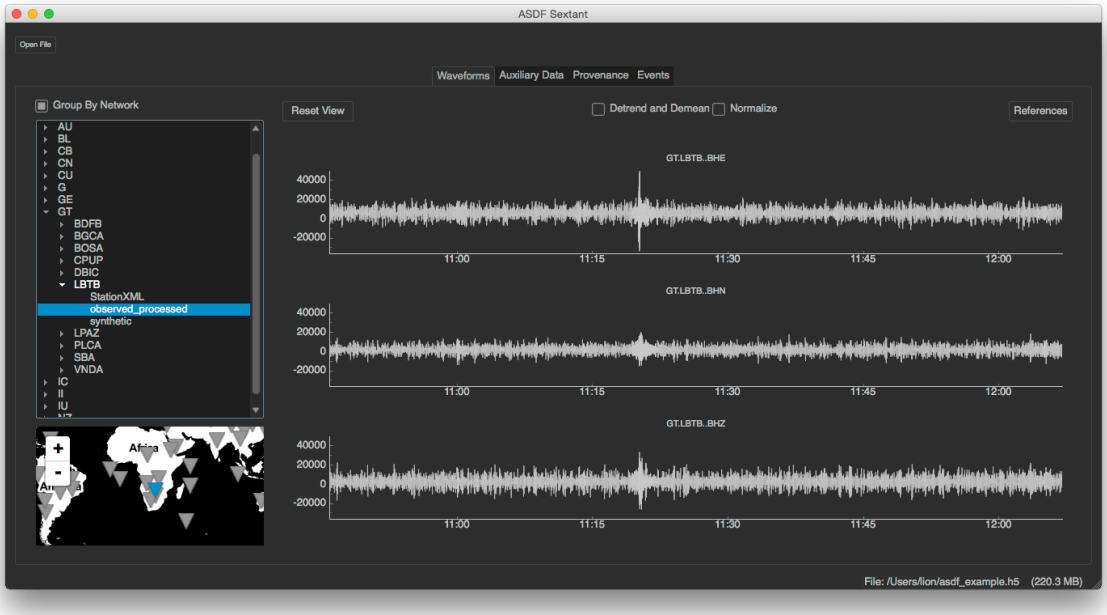

Figure 4.4.: Screenshot of a graphical user interface to interactively explore the contents of an ASDF file. On the left side a recording station can be selected in a tree structure that sorts all stations by network code. For each station the hierarchical waveform tag as well the StationXML document for that station can be selected. If the StationXML document is selected, a plot with the instrument response will be shown, which is not depicted here. If a waveform tag is selected, all waveform data for that tag will be shown in the main panel, where it can be explored in detail. Note the map in the bottom left corner, which highlights the currently selected station. The coordinates are extracted from StationXML documents stored in the ASDF files. The tabs along the top allow the users to explore other aspects of an ASDF file, auxiliary data, provenance documents, and events. This is a concrete example of a tool that can be developed around the ASDF format, and other data visualizations can be readily imagined.

4.4.3 Graphical User Interface

An organized data structure uniting all kinds of information facilitates the development of novel visualizations and tools that integrate all these different pieces to help scientists interpret their data and results. An example of this is shown in figure 4.4, which portrays a graphical user interface to interactively and visually explore the contents of a single ASDF file. More exhaustive and use-case specific tools and interfaces might be developed by various members of the community in the future.

4.5 Demonstrations and Use Cases

ASDF's success will revolve around its adoption by the seismological community. The format can be used in all branches of seismology and also in related disciplines. This section aims to highlight some practical use cases of the ASDF format, emphasizing various features to give seismologists of all trades a foothold into the format. In particular, we highlight some clear and tangible advantages in comparison to existing formats, and we demonstrate the reduced friction of working with data.

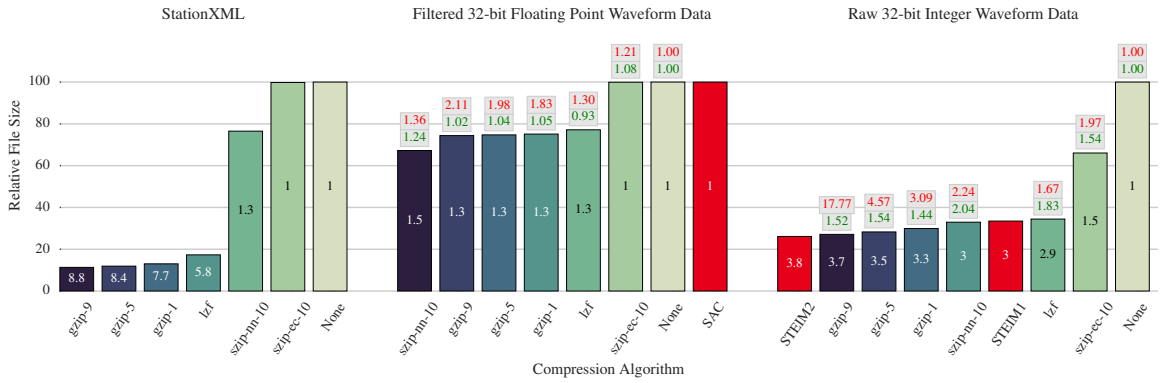

Figure 4.5.: Compression efficiency of the ASDF format using algorithms available in HDF5 for a number of typical seismological datasets. Please keep in mind that the efficiency and I/O speed of these algorithms are heavily dependent on the actual data and hardware, and thus your mileage may vary. The columns represent the file size relative to the uncompressed case, the numbers inside are the achieved compression ratios. The small boxes above the columns denote the relative writing duration in red and the relative reading duration in green compared to the uncompressed case. The left plot shows the efficiency for a dataset containing 500 StationXML documents adding up to 120 MiB. I/O speed differences are irrelevant as the cost for parsing and generation of the XML documents is constant and dominates the total run time. The middle plot shows the compression efficiency for a bandpass filtered waveform dataset stored as 32-bit floating point numbers. It consists of 3,466 waveform traces taking up 282 MiB on disk. The red bar compares it to the uncompressed SAC format. The rightmost plot shows the efficiency for storing 3,346 raw waveform files stored as 32-bit integers taking up 2,340 MiB. The red bars here show the efficiency for the same dataset of the STEIM1 and STEIM2 special purpose compression algorithms defined for the SEED format measured by writing them as MiniSEED files.

4.5.1 Dataset Building

The most obvious use case of ASDF is the creation of full seismological datasets. In this context, a dataset is the collection of all data necessary for a particular purpose. Examples include waveform data for a number of stations for a particular earthquake, all waveforms from a single array, or data from an active source study. The dataset in that sense also contains information about receivers and earthquakes or other sources. In short, the dataset should contain everything that is needed for a certain task. Thus, one no longer needs to deal with complicated and custom directory structures, and it becomes a lot simpler to find required information when it is needed. An additional advantage is that tools and scripts written to work on larger datasets can work on a defined structure, and thus be exchanged and adapted to new uses more easily.

Data organization becomes orders of magnitude simpler when everything is stored in a single file. An added benefit of this is that it decreases the number of files on disk, which, in the case of a large number of files, results in faster transfer and copy times and less issues with file count limits, which are still common on many clusters. The usage of HDF5 also readily grants access to a number of different compression algorithms. Figure 4.5 shows the efficiency and cost of these for a number of typical seismological datasets.

The tools we created and introduced in the previous section are able to convert ASDF files to and from any common seismic file format, so it is very easy to migrate existing datasets.

4.5.2 (Parallel) Large Scale Data Processing

Data processing is something every seismologists must do frequently and has thus been a major focus point in the development of the ASDF format. Data volumes are constantly growing, and

we generally have access to the computing power needed to process and work with it. However, we are at a point where I/O itself, i.e., reading and writing from and to disk, is one of the most expensive parts of many operations.

The data analysis and web industries have many potential solutions to this problem, as manifested in products such as Hadoop or Spark (Apache Software Foundation, 2016a,b). These are very powerful tools which will undoubtedly see successful applications in seismology. A big hindrance with these instruments is that they require significant infrastructure that must be set up and maintained, and their data and workflow models are not a perfect fit to the way most researchers think. Additionally, this field is still in a state of strong flux, and new tools are constantly arriving at the scene.

The HDF5 container being used for the ASDF format enables very efficient parallel I/O by utilizing MPI on machines that support it. MPI is not necessarily any less complex than the previously mentioned tools, but it is one many of us are used to working with, and it is usually installed in our environments. The tools we developed for working with the ASDF format enable scientists to easily construct their own fully parallel workflows, which, for purely practical reasons, they might not have been able to do otherwise. If MPI is not available, our tools fall back on shared-memory parallel execution, which also speeds up executions on a common laptop. Another factor is the great reduction in the number of files, which can result in big gains in speed. Figure 4.6 conceptually illustrates parallel I/O within the ASDF format.

Applications for this operational approach are numerous. Iterative methods might require routine processing of large amounts of data, and speeding this up decreases the time in which research results are obtained. Time critical early warning systems also benefit from reduced latency. A final factor that is not to be neglected is that speeding calculations up enables scientists to keep working on the same problem so they don't need to mentally switch to a new problem while they are waiting for results.

4.5.3 Storage and Exchange of Processed Waveforms

As previously mentioned, many workflows start out by downloading and processing data to bring it into a shape that is suitable for subsequent analysis. While it is occasionally pointed out that one should only store store raw and unmodified data and process it on the fly whenever it is needed, it is oftentimes more convenient and conceptually simpler to process everything once and upfront.

Many research groups create huge processed databases, which keep expanding over a number of years. A big problem is that, in retrospect, it sometimes is very hard to tell what exactly has been stored. The best practices of how to process might have changed, software might have removed or introduced bugs, or the meta information might have changed. If one is not highly diligent, it will not be known what exact software version and which meta data has been used to arrive at a particular piece of data. This creates fidelity problems with the final result of some research topic if one cannot trust nor check the correctness of the initial data that went into it. The situation only worsens when data is exchanged with others, to the point that most researchers will reprocess data with their own true and trusted methods.

ASDF provides a way out of this dilemma by storing processed waveforms with meta information and, most importantly, provenance. As illustrated in figure 4.3, provenance tracks the exact parameters that went into the processing as well as the software and the precise version

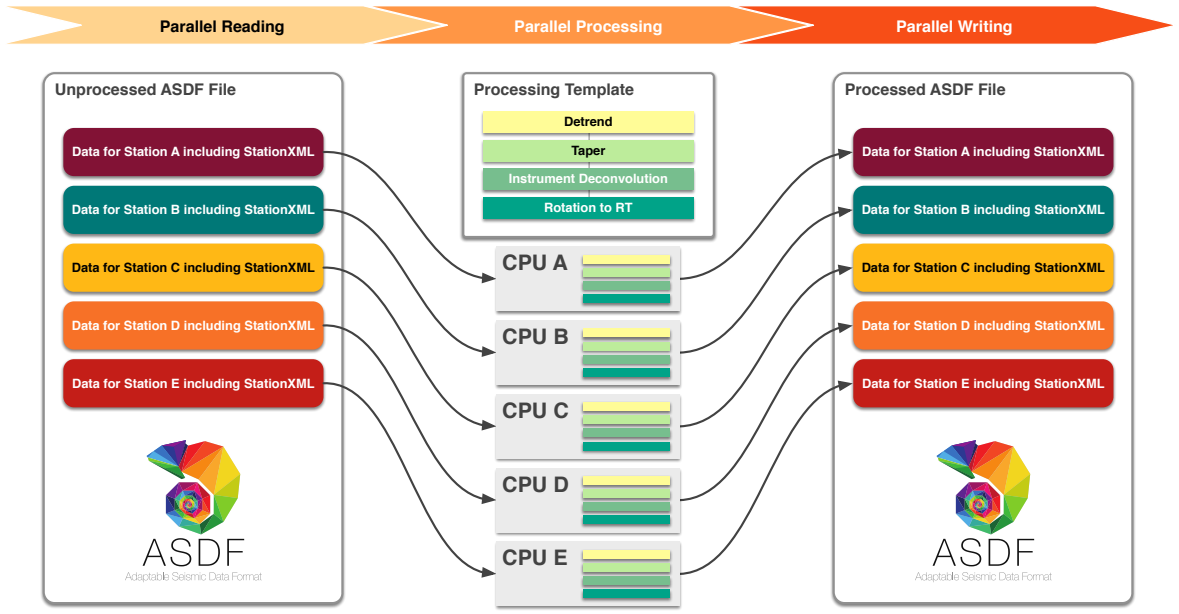

Figure 4.6.: Schematic illustration of parallel data processing with the ASDF format. It illustrates how a very common chain of signal processing operators is applied to all data within an ASDF file producing a new, processed ASDF file. On the left is a file containing unprocessed waveform data including station meta information. We then define a processing template that should be applied to all the data: detrend the data, taper, remove the instrument response, and finally rotate to obtain the radial and transverse components. Note that the last two steps require station information (coordinates and instrument response) and the final one also requires the event coordinates. All the necessary information is readily available in ASDF files. Data from all stations is read in parallel from the file, the template is applied on different CPUs, and finally the data is written in parallel to a new, processed ASDF file. On the proper hardware and if run under MPI the I/O is fully parallel. The real world situation is a bit more complex as it has to deal with memory limitations, different numbers of CPUs and stations, and other complications, but conceptually it is very similar. We provide an implementation that makes this kind of task very easy to perform.

of it that was used for any task. It is stored in the same file as the actual data, and thus it cannot get lost and will always be available.

4.5.4 Storage and Exchange of Synthetic Waveforms

Synthetic seismograms are at the very core of many seismological disciplines, and a slew of analytical, approximate, and numerical methods exist to compute them. While some of these methods are very fast, the computation of accurate waveforms in three-dimensional media with realistic rheologies still requires major computational resources. This makes it worthwhile to attempt to reuse these synthetics in different studies and by different research groups, which might not have access to the aforementioned resources. As source parameters and Earth's internal structure are, at best, imperfectly known, and various approximations might have to be employed in the course of the calculations, the resulting synthetics are highly dependent on the solver and the source and structural models. Thus, this information must be known and communicated to others (which might well be the person who originally performed the simulation), which is a non-trivial issue, to the point where synthetic seismograms are oftentimes only used for one study. One way to resolve this issue is illustrated by the ShakeMovie project (`http://global.shakemovie.princeton.edu/`, Tromp et al., 2010), where a web publication details the used sources and Earth models, and the files needed to steer the waveform solver can be downloaded. All the necessary information is provided if one is willing to look for it, but the amount of documentation and infrastructure this requires is clearly not practical in many other cases.

ASDF offers a suitable way of storing synthetic waveforms, including full provenance information, which was a major motivation of its development. In most scenarios the result is a single file per simulation storing all waveforms for a certain source as laid out in the previous sections, including receiver information and source parameters. Data relations take care to define which precise origin and focal mechanism parameters were used to calculate the seismograms, detailed information about the used solver, Earth model, and input files are part of the provenance, as illustrated in figure 4.7.

ASDF output should be directly integrated into the program calculating the synthetics, as all the information is known only at this particular point in the workflow. This greatly reduces the friction and barrier to entry. Very important side benefits are potential performance gains due to efficient I/O and a reduction of the number of files. Synthetic studies can result in a huge number of seismograms previously requiring one file per station and component, which is cumbersome to work with and strains the file system. We implemented ASDF I/O for the spectral-element solver SPECFEM3D_GLOBE.

4.5.5 Adjoint Tomography Workflow

Seismic tomography acts as an ideal use case for the ASDF data format. First, the data volumes involved are massive, easily containing millions of seismograms. Second, it necessitates sophisticated processing to turn raw data into meaningful results. Here, we present a typical data processing workflow occurring in full seismic waveform inversions with adjoint techniques (Tromp et al., 2005; Fichtner et al., 2006; Tape et al., 2010; Zhu et al., 2012; Colli et al., 2013) but the general idea also translates to other types of tomography (see Liu and Gu (2012) for a recent review).

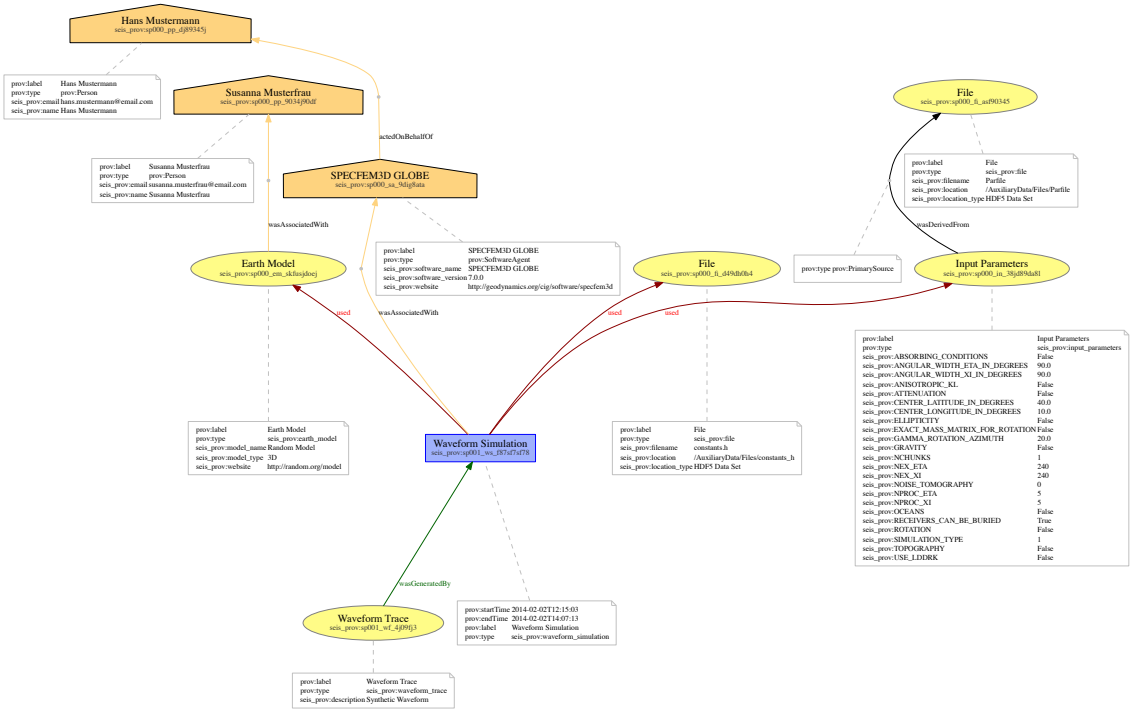

Figure 4.7.: Example of a SEIS-PROV provenance graph for a waveform simulation performed with SPECFEM3D_GLOBE. The final result is the waveform trace entity at the bottom, which was generated by a waveform simulation activity. The waveform simulation in turn used a certain Earth model, and some input parameters and files. It also knows which software was used, including the exact version number. Additionally, one can specify who performed the simulation and who is responsible for a certain Earth model. The aim of the provenance description is to capture all settings and variables that have an effect on the final waveform. The details are not vital for the purpose of this graph – please consult a digital version of this manuscript for a more legible version.

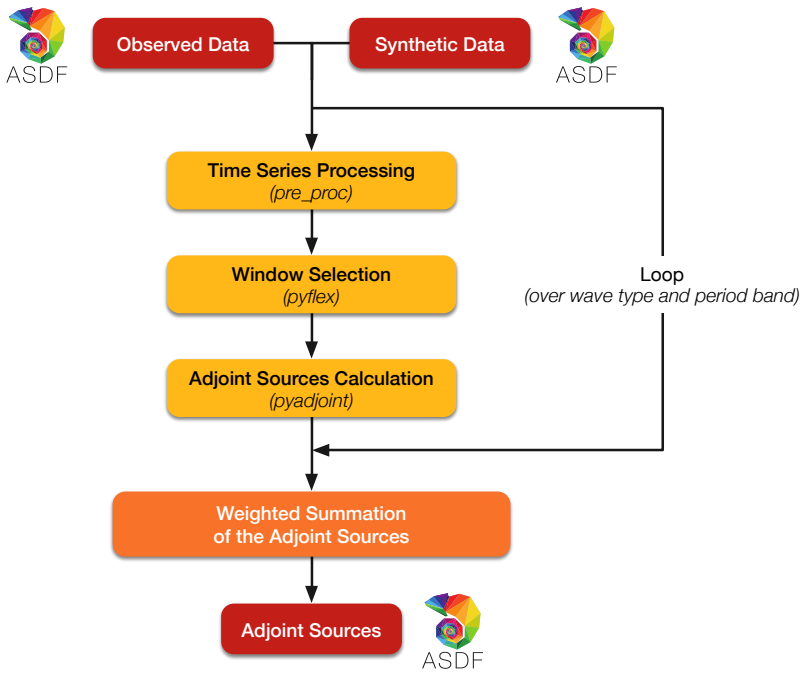

Figure 4.8.: Adjoint tomography preprocessing workflow for a single event with, typically, several thousand stations. Observed and synthetic data are initially available as two ASDF files. These are processed, windows are selected, and adjoint sources are constructed. All of these operations are accelerated with MPI utilizing parallel I/O. This is repeated for each wave type and associated period band, and the final adjoint sources are weighted and stored in an output ASDF file. Note that only three ASDF files are involved; previous iterations of this workflow without ASDF used tens of thousands of files per event. The open-source Python packages pre_proc, pyflex, and pyadjoint collectively comprise a cohesive workflow.

To enable a physically meaningful comparison between observed and synthetic waveforms, they need to be converted to the same units and filtered in a way that ensures a comparable spectral content. This includes standard processing steps like detrending, tapering, filtering, interpolating, deconvolving the instrument response, and others. Subsequently, time windows in which both waveforms are sufficiently similar are selected and adjoint sources are constructed from the data within these windows, see figure 4.8 for a graphical overview.

The following is an account of our experiences and compares a legacy workflow to one utilizing the ASDF format, demonstrating its clear advantages: Existing processing tools oftentimes work on pairs of SAC files, observed and synthetic seismic data for the same component and station, and loop over all seismic records associated with any given earthquake. Given the large number of seismic receivers and earthquakes, the frequent read and write operations on a very large number of single files create severe I/O bottlenecks on modern computing platforms. The implementation centered around ASDF shows superior scalability for applications on high-performance computers: Observed and synthetic seismograms of a single event are stored in only two ASDF files, resulting in a significantly reduced I/O load. What is more, it is beneficial to keep meta information in the same file. For example, one does not need to reach out for separate files that keep track of the stations' instrument information or files containing earthquake information, which greatly reduces the complexity of operations and the possibility of making mistakes. Last but not least, provenance information is kept to increase reproducibility and for future reference.

Other than the data format itself, the workflow benefits from the extensive APIs provided by ASDF. ASDF is, as mentioned, supported in the SPECFEM3D_GLOBE package (Komatitsch and Tromp, 2002a,b): Synthetic ASDF files are directly generated, meaning synthetic data can seamlessly be fed as an input into the workflow. To maximize performance, we rewrote our existing processing tools. A big drawback in the old version was that codes were written in different languages and unable to communicate with each other easily. For example, the SAC package (Helffrich et al., 2013) was used for signal processing and the Fortran based FLEXWIN program (Maggi et al., 2009) for the window selection. In the new version we treat tasks as individual components in a single cohesive workflow. Relying on ObsPy (Beyreuther et al., 2010) and other packages, we re-developed all workflow components in Python. Therefore, all components integrate with each other and stream data from one unit to the next. I/O only happens at the very beginning, when we read the seismogram into memory, and at the very end, when we write out the adjoint sources. All in all these changes empower us to increase the scale of our inversions —in terms of frequency content, numer of earthquakes, and number of stations— and fully exploit modern computational platforms.

4.5.6 Ambient Noise Cross-Correlations

The extraction of information from recordings of ambient seismic noise is a prime candidate for fully utilizing ASDF. For one, the required data volumes are amongst the biggest in our science. Extracting useful information necessitates the correlation of very long time series, which in turn imposes great demands on the I/O system and data organization. ASDF enables the storage of arbitrarily long waveform traces in a single file with fine grained access. One example is storing a station's data for several years in one file and only accessing a portion of the data whenever it is needed. That allows the design of very versatile processing pipelines with a flexible choice of parameters, be it for windowing or other operations.

Ambient noise analysis, and especially its preliminary data processing, is still in heavy flux, and each research group uses their own set of tools and algorithms. That might make it very hard to figure out what processing has been applied to data before it was correlated, which in turn obstructs the discovery of the influence and importance of the various parameters. Provenance greatly improves upon that situation.

Additionally, there is currently no community accepted standard to store and exchange cross correlations and stacks thereof. As their calculation can be very expensive, storage and exchange is a worthwhile endeavor and will facility a wide range of studies that can commence from already calculated correlations. A number of projects to generate and offer databases of correlations are currently being explored or executed. ASDF aims to be the format for that purpose. Its auxiliary data capability and advanced provenance description make it very versatile and suitable for this purpose. We encourage the noise community to adopt the format and settle on a convention for storing cross correlations and their stacks in ASDF's auxiliary data section. At that point it is conceivable that a future generation of the format contains a comprehensive definition of how to store correlations and other products of noise analyses. This also fascillitates an easy exchange of workflows, tools, and human knowledge and skill.

4.5.7 Industry Dataset

Industry datasets are not the primary focus of the ASDF format, but since these datasets share many similarities to passive-source datasets, it is worthwhile proposing how we could adapt ASDF to that particular use case. Such a dataset generally consists of multiple sets of sources and receivers. As previously mentioned, the industry lacks standards for the characterization of sources and receivers, so we took the simple approach of storing the minimal information needed for general data processing tasks with industry datasets. As in the passive-source case, the sources are stored in a QuakeML document and the receivers are stored in StationXML documents.

Sources used in industry data are generally active (for example a bungee-assisted weight-drop): approximate source time functions (extracted by the data processor) and the source location information are stored inside the QuakeML document. Unlike earthquake data, the source time function must be manually determined from recorded data near the source (there is not a catalog of sources for events). Since the same source is often used throughout an active-source array, there is a tag for each source time function that stores the source layout for simple geometries. The storage of the source-time function also allows simulations to be run with this source-time function in waveform solvers, such as SPECFEM3D.

The StationXML documents store information about each instrument, including its response (for example as poles and zeros), basic metadata (gain, amplification factor, etc), and instrument locations. Since these instruments are often layed out in an array configuration, there is a tag for each instrument that stores the geophone layout for simple geometries inside the StationXML documents.

Waveforms are grouped by the recording instrument. One network corresponds to one receiver layout. This is in contrast to the SEED naming convention that is used to organize the data in the passive-source case. Each network can have multiple geophone types — figure 4.9 illustrates this concept. The result is perhaps a bit less clean than the passive-source case, but it demonstrates ASDF's adaptability to adjust to a large number of use cases.

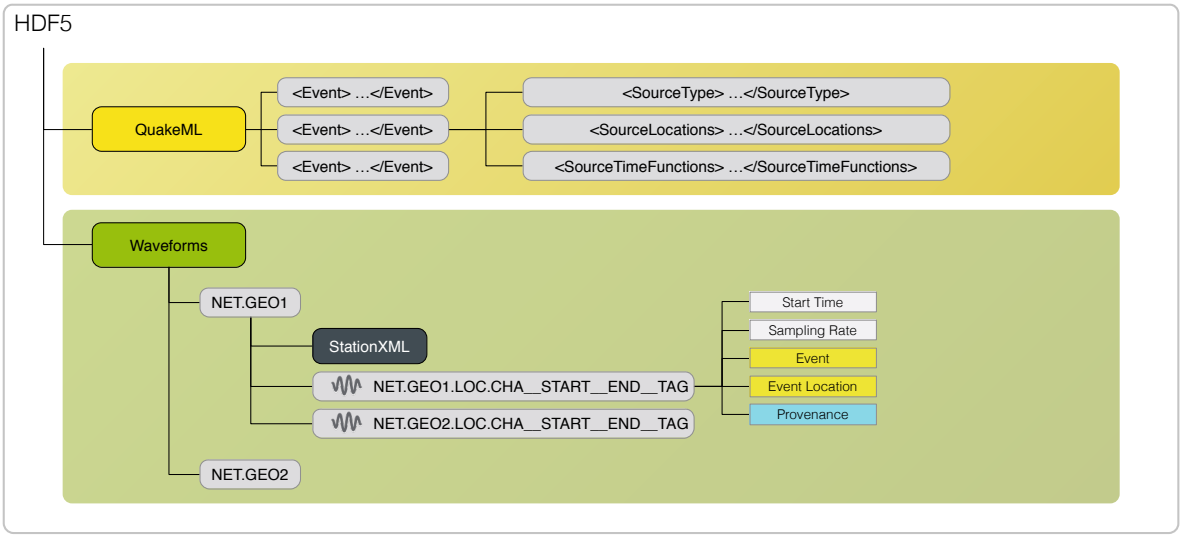

Figure 4.9.: The ASDF container adapted to an active-source dataset. This is a specialized version of figure 4.2 — the auxiliary data and provenance sections are omitted for the sake of brevity but are still present in the file. The figure demonstrates how ASDF can be adapted to various use cases. In this particular case, the QuakeML document is slightly augmented as is the internal grouping. See the main text for details.

The ASDF container is compatible with ObsPy, giving researchers access to powerful signal processing tools for their data. We believe the industry would benefit from adopting ASDF, since the format offers improved data organization, simple but efficient parallel processing, and provenance capabilities all wrapped up in a modern format.

4.5.8 Further Uses

Many more use cases of the ASDF format can be envisioned, and we hope different subgroups within the seismological community will adapt it for their own purposes. Aside from seismological waveforms, ASDF's ability to save auxiliary data, including full provenance, enables it to store a lot of different pieces of data.

Examples include storing time dependent power spectral densities and combining them into probabilistic power spectral densities on the fly (e.g., McNamara and Buland, 2004) or building a database of historical earthquake data. Even non-seismological data, such as GPS time series and magnetotelluric data, is not out of the question and would benefit from the provenance description and the advanced processing tools developed around ASDF. Some of these examples are already being attempted, and we intend to maintain a collection of use cases on our website.

4.6 Conclusion

ASDF has been developed with the broader seismological community in mind, and our hope is that scientists within this community will continually test, offer feedback, and improve ASDF and its associated tools. Through such a communal effort, we will gracefully meet future data challenges and empower ourselves to make new scientific discoveries.

Figure 4.10.: The left box is a screenshot of the `http://seismic-data.org` website, the central part shows the ASDF logo, and the right box is a screenshot of an article which is part of the pyasdf documentation. A lot of effort has been put into these sites in an attempt to build a community around the format and gather a critical mass of users.

All components of the format, including its definition, implementation, and other tools, are freely available under open source licenses and hosted on GitHub. A central entry point to the ASDF format is the `http://seismic-data.org` website, which, amongst other things, is shown in figure 4.10. We welcome any outside comments, criticisms, and success stories, and we are committed to maintain the documentation and implementations for the foreseeable future.

Acknowledgements

This research was partially supported by the EU-FP7 VERCE project (number 283543) and US NSF grant 1112906. We are grateful for the QUEST Initial Training Network (Marie Curie Actions, `http://www.quest-itn.org`) and the Computational Infrastructure for Geodynamics (CIG, `https://geodynamics.org/`) organization for holding a joint workshop that sparked the creation of the ASDF format.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

The authors also recognize support from the NSERC G8 Research Councils Initiative on Multilateral Research Funding and the Discovery Grant No. 487237.

Additionally we thank Chad Trabant and Tim Ahern from the Incorporated Research Institutions for Seismology (IRIS) as well as Emiliano Russo, Peter Danecek, and Rodolfo Puglia for fruitful discussions and useful tips. Finally, we gratefully acknowledge conversations with HDF5 Director of Earth Science Ted Habermann and help from Mohamad Chaarawi via the HDF5 User's Forum.

5
North America Inversion

Using the technical advances introduced in the previous chapters, we present a new seismic tomography model based on a full seismic waveform inversion for crustal and upper mantle structure from the western edge of North America across the Northern Atlantic well into Europe. The initial model has been extracted from the Collaborative Seismic Earth Model (CSEM; Afanasiev et al., 2016) and contains contributions from the global S20RTS model (Ritsema et al., 1999), as well as two regional full waveform models (Fichtner et al., 2013; Colli et al., 2013). The final results of the inversion are flowing back into the CSEM in an ongoing effort to image our whole planet across the scales.

The actual inversion strategy utilizes the adjoint-state method coupled with an L-BFGS quasi-Newton optimization scheme. To assist convergence towards the global minimum of the misfit given our data, we break down the inversion into several steps. Starting with inverting long period data to image the large-scale structure, we successively increase the frequencies, to resolve the finer details.

This highly iterative and error-prone process is fully automated across all stages, from data downloading and quality control, over window picking, running forward and adjoint waveform propagation simulations, to misfit calculations and the computation of the search directions during the numerical optimization.

Significant challenges include the size of the domain, the unbalanced data coverage, and the strong east-west alignment of seismic ray paths across the North Atlantic. We use as much data as feasible, resulting in several thousand recordings per event, depending on the receivers deployed at the earthquakes' origin times. Our automated process ensures a reproducible and trustworthy result.

The work in this chapter has been carried out in collaboration with Andreas Fichtner, Christian Boehm, and Heiner Igel.

5.1 Introduction

Inversions for seismic velocities of the subsurface using recorded seismic data face the difficult task of using point-wise surface measurements in the form of recorded seismograms to indirectly infer information about volumetric quantities, e.g. the elastic parameters of the Earth. An uneven coverage of these surface measurements enlarges this impediment: Densely populated areas at risk in developed countries are heavily instrumented whereas large parts of the planet (e.g. oceans and deserts) are blind spots in terms of seismic recordings, though efforts to change this are underway (Sukhovich et al., 2015). Furthermore, data, in addition to the measured waveforms of interest, contains environmental and instrument noise, potentially strong effects of the local subsurface, and plain errors arising from technical or operational issues which obstruct or alter the measured quantity. Having more data has the potential to alleviate both these problems by (a) covering previously uncovered areas and (b) statistical averaging to lessen the relative importance of each recording and thus the sensitivity to errors and certain local effects like timing errors of seismic receivers, for example.

The seminal USArray project (`http://www.usarray.org`, last accessed November 2016) offers an, at continental-scale, unprecedentedly dense and high quality seismic sensor network for the continental United States of America. Utilizing all that data is a logical next step for full seismic waveform inversions but technical roadblocks made that difficult and error prone. The advances detailed in the previous chapters in the form of the ObsPy toolkit (Beyreuther et al. (2010); Krischer et al. (2015b); chapter 2), the LASIF framework (Krischer et al. (2015a); chapter 3), and the ASDF data format (Krischer et al. (2016); chapter 4) enable us to tackle that problem.

First seismic tomographies appeared in the late 1970s (Dziewoński et al., 1977; Aki et al., 1977) using time of flight (or travel time) data. More data, coupled with theoretical developments, led to ever more sophisticated models (Kissling, 1988; Spakman, 1991; Yomogida, 1992; Grand et al., 1997; Dahlen et al., 2000; Rawlinson and Sambridge, 2003; Friederich, 2003; Yoshizawa and Kennett, 2004, 2005; Sigloch et al., 2008) resulting in today's state-of-the-art inversions using full waveforms with 3-D numerical forward and adjoint simulations (Tarantola, 1988; Tromp et al., 2005; Fichtner et al., 2006, 2009; Tape et al., 2010; Zhu et al., 2012; Bozdağ et al., 2016); see Liu and Gu (2012) for a recent review.

The chapter at hand introduces a new full seismic waveform inversion model with the explicit goal to harness the data produced by USArray. Figure 5.1 shows the chosen study region: It contains the complete continental United States and has been expanded to the West, South, and East in order to include tectonic boundaries granting access to data from more large earthquakes. Moreover, the study region contains a significant part of western Europe to take advantage of the dense instrumentation in that region. Diverse tectonic settings from the subducting plates beneath the west coast of North America, over the complicated situation in the Caribbean Sea, to the Mid-Atlantic Ridge, including the hotspot beneath Iceland, provide ample reasons to study this domain in as much detail as possible.

The rest of this chapter is structured as follows: Section 5.2 details the chosen forward and inverse modelling scheme, followed by our approach to meet the encountered workflow challenges in section 5.3 and an overview of the used data in section 5.4. Results and resolution proxies are finally presented in section 5.5, with potential next steps being discussed in section 5.6.

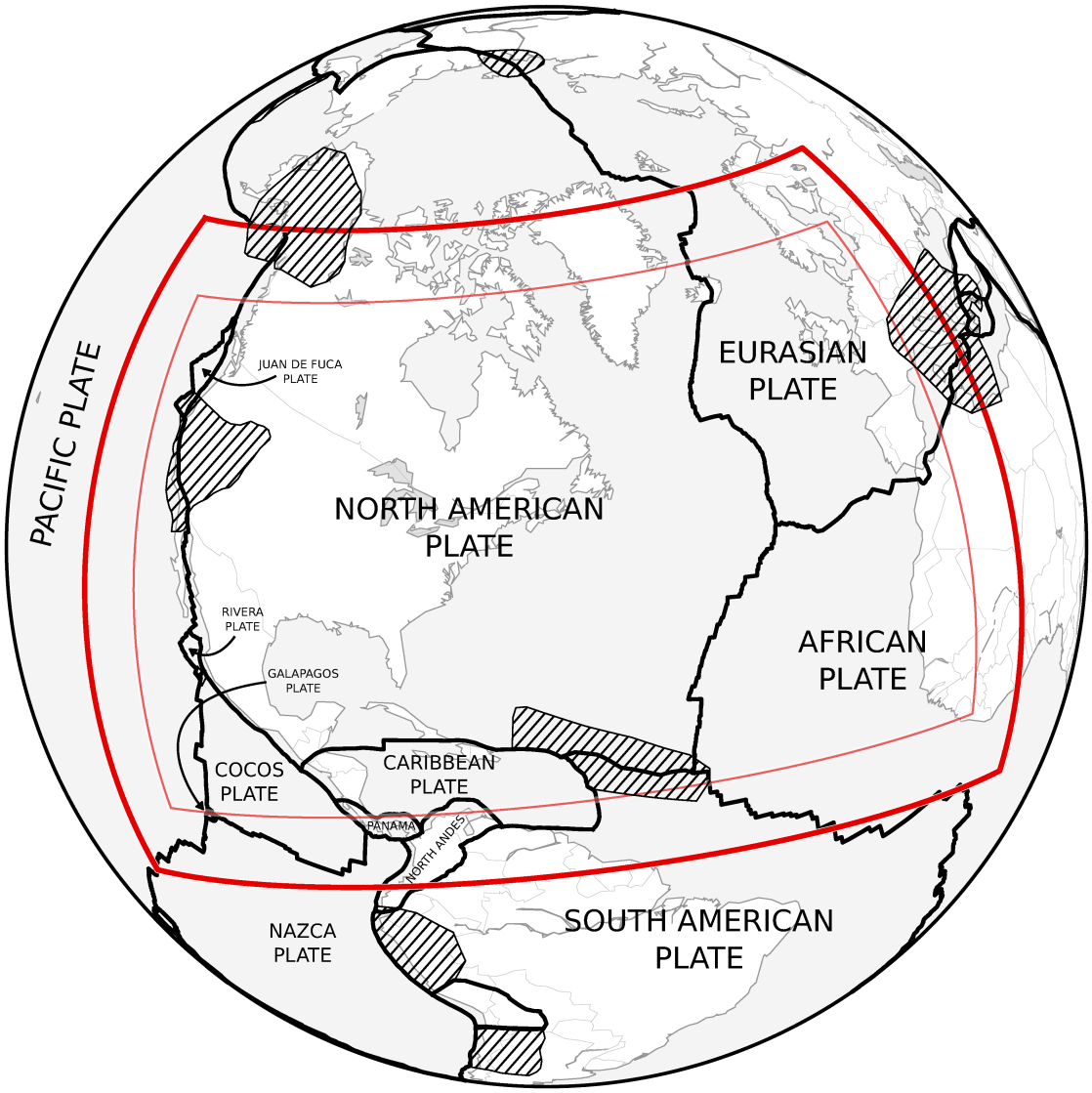

Figure 5.1.: The chosen physical inversion domain and its tectonic setting. The outer red boundary marks the full domain in which we ran our simulations and inversion procedure. In order to mitigate the effects of the unphysical boundaries on the inversion we only invert for data whose direct raypaths are within the inner red boundary. The black lines show the tectonic plate boundaries and the hatched areas are orogens where a further microzonation in smaller tectonic plates is not performed. Plate boundaries and orogens are from Bird (2003).

5.2 Forward and Inverse Modelling

This study aims to develop a seismic wave velocity model in the crust and upper mantle beneath our domain of interest. As direct measurements are impossible, other means have to be employed. The field of seismic inversions studies the relation between observed data and their theoretical or numerical predictions to infer Earth's structure. Full seismic waveform inversion in particular strives to use all parts of seismograms where a physically meaningful misfit measurement between synthetic and observed data can be computed. Our applied scheme compares recorded to numerically calculated seismograms and uses the difference between both to iteratively improve the model. By and large, we follow the methodology established in Fichtner et al. (2009); Fichtner (2011); Tape et al. (2010) and similar works, but we scale it up to more data and deviate in several aspects, which are detailed in the following.

5.2.1 Waveform Modelling and Starting Model

Solutions of the elastic wave equation and its adjoint-state with a viscoelastic rheology in a radially anisotropic medium are computed by SES3D (Gokhberg and Fichtner, 2016). It employs the spectral element method (Seriani et al., 1995; Faccioli et al., 1997; Komatitsch, 1997) which, aside from its computational efficiency, has natural free surface boundary conditions, a mandatory trait for modelling surface wave dominated data. Long period physical effects like gravitational influences of propagating waves are not simulated, limiting the upper fully usable period of the calculations to about 120 seconds.

The employed gradient-based optimization scheme mandates a good starting model, which has been extracted from the Collaborative Seismic Earth Model (CSEM) project (Afanasiev et al., 2016). It is made up of a global 1-D background model based on PREM (Dziewonski and Anderson, 1981) including its attenuation model, superimposed by 3-D perturbations to the S-wave velocity from S20RTS (Ritsema et al., 1999), its perturbations to the P-wave velocity are scaled to it. Globally, the crust is derived from a model by Meier et al. (2007). The specific inversion domain of this chapter additionally contains contributions from a full seismic waveform model of Europe (Fichtner et al., 2013) as well as the Southern Atlantic (Colli et al., 2013). These can be clearly seen in the initial model in figure 5.2. The final result of this work flows back into the CSEM so it can serve as a foundation for future generations of Earth models.

5.2.2 Optimization Scheme

Our wave propagator uses models parameterized in vertically and horizontally propagating P and polarized S wave speeds v_{PH}, v_{PV}, v_{SH}, v_{SV}, the density ρ, shear attenuation Q_μ, and the dimensionless parameter η (see Dziewonski and Anderson (1981) for more details) which controls the incidence angle dependent propagation speed of seismic waves. Independently constraining all these parameters is not realistic with the given data coverage, thus, we invert only for the isotropic P wave speed v_P (forcing v_{PH} and v_{PV} to be equal), v_{SH}, v_{SV}, and ρ to reduce the parameter space. η is forced to 1. See Fichtner et al. (2013) for a more detailed explanation and reasoning.

Inversions for density anomalies are usually poorly constrained as their effect on the seismic wavefield is comparatively small. Many past works thus chose not to invert for density, but it

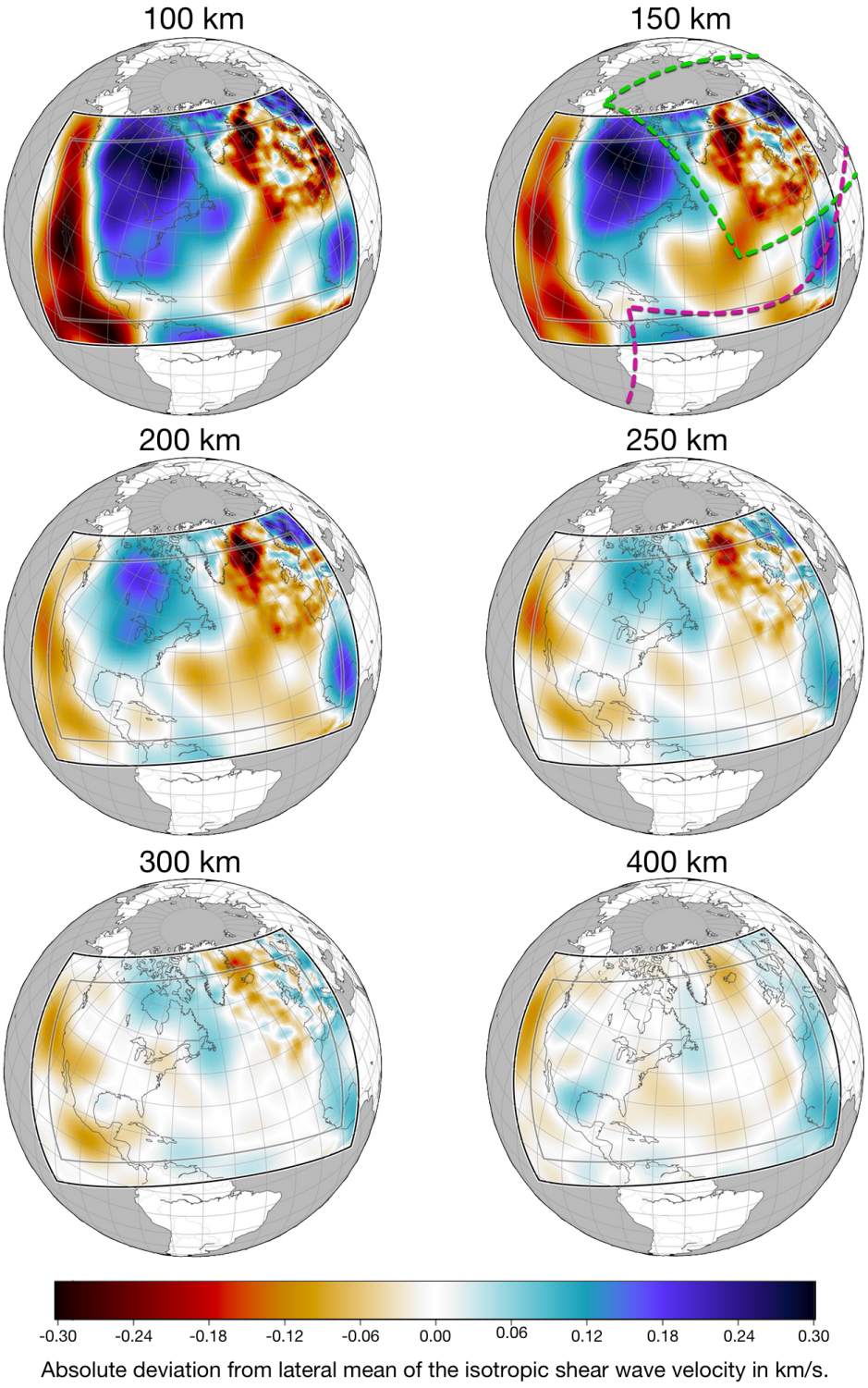

Figure 5.2.: Horizontal slices through the initial model showing the isotropic shear wave speed at various depths with the same color scale. The slice at a depth of 150 km additionally shows the two existing full waveform inversion models that are part of the initial model: The green dashed border is an outline of a model for Europe from Fichtner et al. (2013) and the purple dashed border denotes the extent of the South Atlantic model from Colli et al. (2013).

can be shown that doing so negatively affects the recovery of the other parameters (Blom et al., 2017). Hence, we include it in the optimization, but do not interpret it.

The model is updated by calculating the gradient of our objective functional with respect to the model parameters using the adjoint-state method (Tromp et al., 2005; Fichtner et al., 2006; Tape et al., 2007). It is a computationally efficient technique for calculating the gradient of a chosen misfit measurement with respect to 3-D velocity models, presuming a large number of seismic receivers per event as calculating the gradient for a specific model realization requires only two numerical simulations (forward and adjoint) per event, independent of the number of stations.

Given an Earth model $\boldsymbol{m}_k$, we aim to find a new Earth model $\boldsymbol{m}_{k+1}$, so that

$$\chi\left(\boldsymbol{m}_{k+1}\right) < \chi\left(\boldsymbol{m}_k\right)$$

for a given objective functional $\chi\left(\boldsymbol{m}_k\right)$ comparing observed to synthetic seismograms calculated with model $\boldsymbol{m}_k$. This will gradually force the model to improve in the sense that synthetic data will become more similar to observed data. The adjoint-state method yields the gradient $\nabla_{\boldsymbol{m}_k}\chi\left(\boldsymbol{m}_k\right)$ of the objective functional for any given Earth model with respect to its model parameters, called $\nabla\chi\left(\boldsymbol{m}_k\right)$ in the following. One possibility to find a new model is to walk into the direction of the negative gradient with step length α:

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k - \alpha\nabla\chi\left(\boldsymbol{m}_k\right).$$

This is the simplest gradient-based method known as steepest or gradient descent and is guaranteed to reduce the misfit for a small α and a continuous and smooth enough misfit surface. This tends to converge fairly slowly and requires a number of test step lengths to discover an acceptable α for each iteration, each of which requires another set of simulations.

In order to alleviate these issues we employ the L-BFGS method (Liu and Nocedal, 1989) which is a low-memory variant of the quasi-Newton BFGS method; see Nocedal and Wright (2006) for an extensive description. Crudely speaking, it approximates the curvature of the misfit functional by approximating the inverse of the Hessian $\nabla^2_{\boldsymbol{m}_k}\chi\left(\boldsymbol{m}_k\right)$, using the gradients and models from successive iterations. The approximate inverse Hessian $H^{-1}\left(\chi\left(\boldsymbol{m}_k\right)\right)$ is then used to perform a quasi-Newton step and update the model with

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k - \alpha H^{-1}\left(\chi\left(\boldsymbol{m}_k\right)\right)\nabla\chi\left(\boldsymbol{m}_k\right)$$

In contrast to first-order gradient-based methods, this has a natural step length of 1 so the first test model is usually directly accepted and can be used as the forward simulation for the next iteration. A direct consequence is that one only needs two sets of simulations per update, greatly speeding up the whole process. In practice this works fairly well but the non-linearity of the problem occasionally requires to lower the step-length; see section 5.5 for failed test models encountered in the course of the inversion. Modrak and Tromp (2016) show that L-BFGS converges faster compared to other optimization schemes.

BFGS methods estimate the current Hessian by inferring curvature information from past iterates. Each step must satisfy the *curvature condition*:

$$\boldsymbol{s}_k^T\boldsymbol{y}_k > 0$$

where $\boldsymbol{s}_k = \boldsymbol{m}_{k+1} - \boldsymbol{m}_k$ and $\boldsymbol{y}_k = \nabla\boldsymbol{m}_{k+1} - \nabla\boldsymbol{m}_k$. This is enforced by requiring each update to satisfy the so called *Wolfe conditions*:

$$1. \qquad \chi\left(\boldsymbol{m}_k + \alpha \boldsymbol{p}\right) \leq \chi\left(\boldsymbol{m}_k\right) + c_1 \alpha \nabla \chi\left(\boldsymbol{m}_k\right)\ \boldsymbol{p}$$

$$2. \qquad \mid \nabla \chi\left(\boldsymbol{m}_k + \alpha \boldsymbol{p}\right)\ \mid \leq c_2 \mid \nabla \chi\left(\boldsymbol{m}_k\right)\ \boldsymbol{p}\ \mid$$

with $\boldsymbol{p}$ being the search direction and $0 < c_1 < c_2 < 1$. The first condition is known as *Armijo's condition* and makes sure each update has a sufficient decrease of the objective function. The second is the so called *curvature condition* and assures that the slope at each iterate is smaller than c_2 times the initial one. Both have to be satisfied for a test step length to be acceptable

In full waveform inversion there is a big difference in both, as Armijo's condition only calls for the calculation of the misfit whereas the curvature condition also needs the gradient of the current test model, demanding twice as many numerical simulations. Rejecting a test model based on the curvature condition is therefore very costly. The literature, e.g. Nocedal and Wright (2006), suggests values for c_2 of about 0.9 for quasi-Newton methods. At later iterations, each successful reduction of the misfit is desirable and we thus set c_2 to 0.95 which manages to avoid a lot of rejected test models.

5.2.3 Misfit Functional

The concrete choice for the objective functional $\chi\left(\boldsymbol{m}\right)$ of the seismograms of a particular model $\boldsymbol{m}$ has a large influence on the outcome of an inversion. It determines if and how observed and synthetic data have to be windowed and what seismic phases can be used. Long-wavelength Earth structure mainly speeds up or slows down seismic waves and using the L^2-distance between observed and synthetic data does not explicitly extract that information. Luo and Schuster (1991) argue that using a cross-correlation time shift misfit functional has a more linear relationship with the model parameters. Tape et al. (2007) extends that to frequency-dependent multitaper (Thomson, 1982) measurements. Instantaneous phase and envelope misfits (Bozdağ et al., 2011; Rickers et al., 2012) are an alternative approach to this problem.

This work chooses a closely related phase misfit measured in the time-frequency domain introduced in Fichtner et al. (2009). It is conceptually very simple by transforming both, observed and synthetic data, to the time frequency domain and performing a phase difference measurement in each point in the domain. A particular advantage is the ability to add weights in the time-frequency domain which enables the exclusion of areas in the time-frequency domain that do not have enough energy to yield a meaningful phase difference calculation. A downside is that it requires careful window selection but this is taken care of by the LASIF framework.

The total phase misfit χ_p is defined with an integral over all phase differences in the time frequency domain as

$$\chi_p^2 = \iint W^2(t,\omega)\left[\phi_{syn}(t,\omega) - \phi_{obs}(t,\omega)\right]^2 dt\, d\omega$$

with W^2 being the weighting function, and ϕ_{syn} and ϕ_{obs} the phase of the synthetic and the observed seismogram trace respectively. For a derivation and the corresponding source to the adjoint-state wavefield please see Fichtner et al. (2009) and Fichtner (2011). Consequences of the choice of misfit functional are discussed in section 5.5.

5.2.4 Multiscale Inversion

The chosen starting model is comparatively smooth in large areas of the inversion domain. Instead of immediately inverting for waveforms with high frequency energy, we start by first inverting for long periods to determine the large-scale structure before going to shorter periods. For one, this greatly reduces the computational cost required to determine the large-scale structure. Directly inverting for short period data can additionally trap the optimization in a local minimum (see Fichtner (2011) for an illustrative example); we attempt to avoid this by dividing the inversion into three legs:

- **Leg A:** Periods from 70 - 120 seconds
- **Leg B:** Periods from 50 - 120 seconds
- **Leg C:** Periods from 30 - 120 seconds

See figure 5.3 for an artistic illustration of the actually used numerical finite element meshes. An L-BFGS optimization has to be restarted each time the definition of the misfit or objective functional changes. Using higher frequency simulations does this, so we also use this chance to introduce more data into the inversion, refer to section 5.4 for more details.

5.2.5 Domain Boundaries and Depth Scaling

Seismic wave propagation is a global phenomenon but we only model it in a limited spatial domain and thus have to be careful to avoid boundary effects. A fairly wide buffer zone, about two to three times the width of the area with active perfectly matched layers (PML) boundaries, is allocated along all domain borders. We discard all data that has direct ray paths crossing that boundary zone and also taper our gradients so they are largely zero in it. This slows down convergence and also makes the computations quite a bit more costly, but manages to avoid contamination of the final model from boundary effects to a certain degree. Possible boundary effects are two-fold but related:

(i) The employed PML scheme of SES3D is not a boundary condition in its strict sense as it modifies the physics of the elements closest to the boundary. Thus, any inverted model in that region would be meaningless and cannot be expected to be similar to the parameters of the true Earth.

(ii) The seismograms between each source-receiver pair are sensitive to model parameters in a fairly wide region around its direct ray paths. If any part of that sensitivity is in the region of the PML elements the seismograms will be affected. Thus the inversion would attempt to match seismograms influenced by different physics.

Last but not least, we apply a depth dependent weighting to the gradients to account for the fact that our gradients are dominated by surface wave measurements and they loose sensitivity with depth. Deeper parts are thus up-weighted a bit. We calculate this weighting once at the beginning of each L-BFGS round.

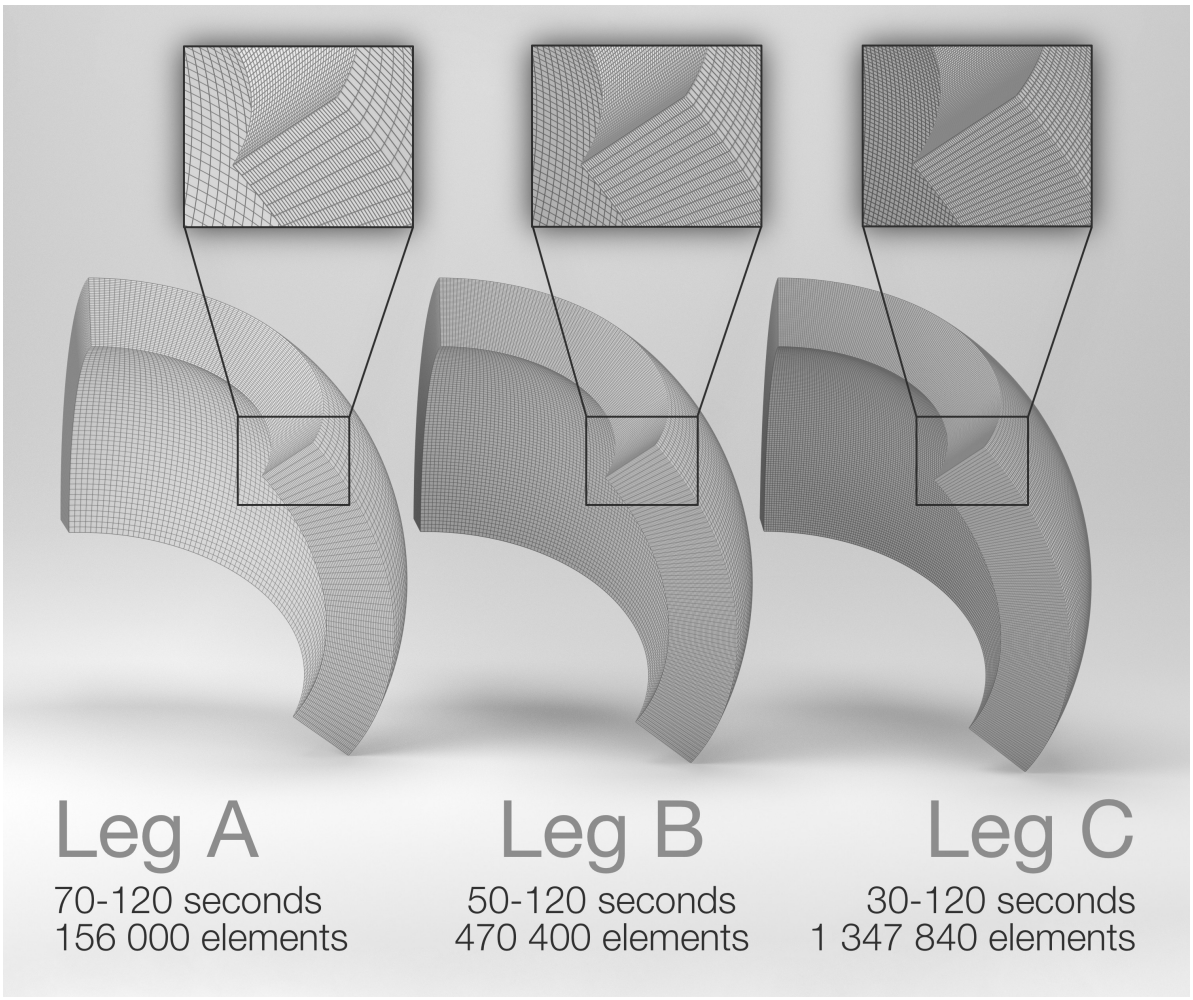

Figure 5.3.: Artistic rendition of the used numerical meshes for the different legs of the inversion. The smaller elements of Leg C compared to Leg A also necessitate a smaller time-step. As a consequence, a single iteration of Leg C is more than 20 times as expensive as a single iteration for Leg A.

5.3 Workflow

Full waveform inversions using adjoints have been carried out for some years (Tromp et al., 2005; Fichtner et al., 2006, 2009; Tape et al., 2010; Zhu et al., 2012; Rickers et al., 2013; Colli et al., 2013; Simute et al., 2016; Bozdağ et al., 2016) and their theory is understood to some extent. A main difficulty in actually carrying out such inversions nowadays lies in the required data management and workflow orchestration which is elaborated upon in this section.

An initial model (see section 5.2.1) is used to calculate synthetic seismograms for a number of events. These seismograms are then compared to observed data in some fashion and their difference is encoded in so called adjoint sources, the source terms of the adjoint wave equation. The subsequent adjoint waveform simulations will yield the local gradient of the chosen misfit functional with respect to the model parameters, which is then used to update the model as explained in section 5.2.2. If the step-length is chosen small enough that update is guaranteed to reduce the misfit if not already at a local minimum. The whole cycle repeats until no more significant reduction of the misfit can be achieved or the model appears to be over-fitting to the data at hand. Figure 5.4 illustrates the process.

That simplified view does not capture the full complexity of the task, as recorded data has to be downloaded, instrument corrected, and filtered to match the spectral content of the synthetics, the simulations have to be set-up, launched, and monitored, one forward and one adjoint simulation per event and per iteration. Suitable windows have to be chosen in which observed and synthetic data are similar enough to enable a physically meaningful comparison. Potentially weighted misfits and adjoint sources have to be derived for each window. The resulting adjoint sources have to be converted to a format the waveform solver can read and stored in a place it can access. Gradients from separate events have to be read and summed up. These and various other tasks have to be dealt with in a reliable and trustworthy manner and the Large-Scale Seismic Inversion Framework (LASIF, Krischer et al. (2015a), chapter 3) mostly takes care of them.

The spectral element method is used for the forward and adjoint simulations for its implicitly satisfied free surface boundary condition and its good scalability, as reasoned in the previous section. Adjoint simulations require access to the wavefield from the forward runs which is oftentimes done by storing the forward wavefield every couple of time steps. For high-frequency simulations and many events the temporary storage costs become prohibitive and on the order of several 100 TB per iteration for this study. To reduce this load we employ a simplified version of the wavefield compression introduced in Boehm et al. (2016) by storing the forward wavefield only at a polynomial degree of 2 (instead of 4 which is used for the simulations) and by using a variable number of bits per element depending on its maximum value. This pushes the total load on the file system per iteration down to about 10 TB for our study which is manageable on modern high performance platforms.

LASIF, by design, does not take care of the actual numerical optimization procedure and delegates that task to an external tool. This optimization toolbox, which will be detailed in a future publication, optimizes with the L-BFGS method (for details see section 5.2.2 and references therein). L-BFGS estimates the Hessian via the gradients of up to five previous iterations and it thus assumes that the given gradients are exact. Numerically calculated gradients always have some inaccuracies but this should not prevent L-BFGS from working, although stronger modifications to the gradients at a certain point will. Gradients from full waveform inversions require some form of regularization due to issues like uneven data coverage, data noise, and sin-

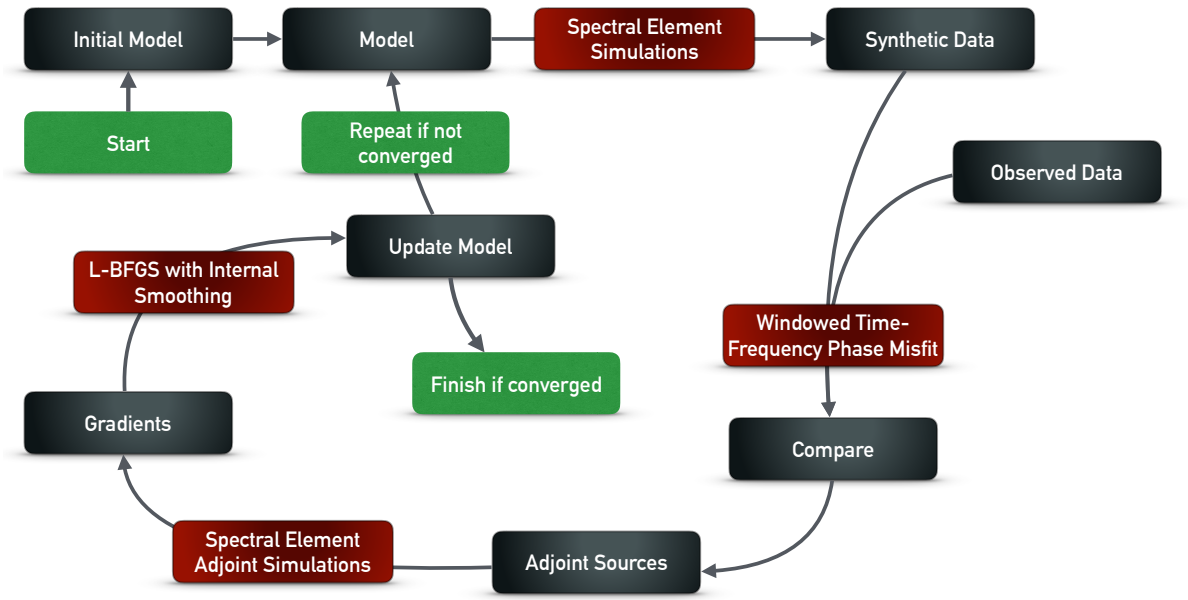

Figure 5.4.: Schematic workflow for full seismic waveform inversions. Starting from an initial model, observed and synthetic data from that model are compared to get the gradient of the misfit with regard to the model parameters. This is then used to update the model to reduce the discrepancy between both. Repeat until converge or the risk of over-fitting becomes too large. Dark gray boxes are the different pieces that enter a full waveform inversion using adjoint techniques, red boxes are implementation details with the choices taken in this study, and green boxes denote the start and end of an iteration. For more details, please refer to the main text.

gularities around the sources and receivers. The chosen cure is usually to smooth the gradients in some fashion, but this, of course, changes the gradients and the indicated descent direction is no longer exact. As L-BFGS uses gradients from multiple past iterations that effect can add up. To circumvent this, we add the applied Gaussian smoothing to the internal parameterization in the L-BFGS algorithm so it is aware of it and can account for its effects. Note that this does not affect direct gradient descent as a smoothed negative gradient still points in a descent direction. The same logic applies to conjugate gradient schemes as the conjugate direction of two smoothed gradients likely is still a reasonable direction, but this should be studied in more detail.

The workflow is harder to execute with L-BFGS, as it needs to be aware of gradients from various iterations at the same time. Doing this by hand proved to be very error prone so we developed a workflow orchestration tool that can react to whatever the optimization toolbox requires at a certain stage. In most cases this is the misfit or the gradient of the seismic model at a certain iteration. The workflow tool reacts to this by calling the proper functionalities of LASIF, launching simulations on high performance machines, copying and merging gradients, calculating adjoint sources and misfits, and returning the results to the optimization toolbox. Some of these operations can take hours to days and everything is thus handled in a fully automated fashion where tasks are executed according to a defined graph structure and dispatched with a job queuing system. The inversion progress can be monitored and controlled via a web based interface depicted in figure 5.5. Details to the workflow tool will be published in a separate publication. Interested readers can contact the authors for access to the tool.

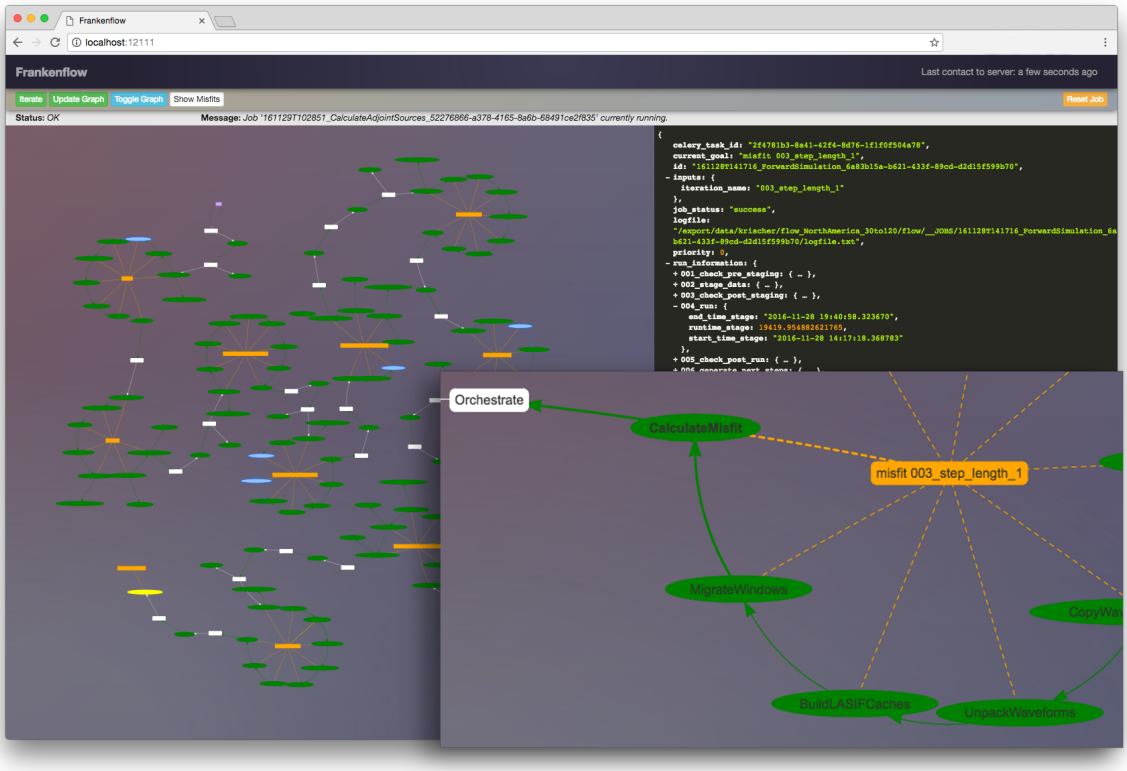

Figure 5.5.: Screenshot and zoomed-in panel of the web based interface to monitor the progress of an inversion in our workflow tool. The right side presents details about a single task like its inputs and outputs and runtime for various internal stages. The larger left panel depicts the graph structure in which the tasks are organized. Each green box is a single task, many of which are grouped around orange goal tasks. The white boxes finally represent orchestration nodes which mainly interact with the optimization toolbox. The currently running task is shown with a yellow ellipse. Everything is fully automated and the user can interact at any point in time by pausing or restarting tasks. Blue ellipses represent tasks where this was necessary.

5.4 Data

An explicit goal of this study has been to use as much data as feasible while still being physically reasonable. Additionally, the simulation cost for forward and adjoint runs does not depend on the number of receivers, thus using the maximum number of receivers is desirable. Following that reasoning we restrict the study to only use earthquakes occurring since the USArray project has been operating with a significant number of stations in 2005.

Observed waveforms need to be clearly recorded at frequencies matching the simulations at stations across the domain. This gives a practical lower limit for the magnitude of usable events of $M_W = 5.0$. The corresponding upper limit of $M_W = 6.5$ was empirically chosen to avoid strong and unmodelled finite source effects. The simulations only use point sources and that approximation therefore has to hold.

The limited maximum magnitude is particularly troublesome for the first, longest period leg of the inversion using periods from 70 to 120 seconds. Events with small magnitudes and thus small rupture surfaces just do not excite a lot of long period energy which, coupled with many instruments lacking sensitivity in these period ranges, results in many unusable recordings, especially on the horizontal components.

All in all, we selected 55 earthquakes from the GCMT catalog (Ekström et al., 2012) for Leg A, and 72 for Legs B and C. For the event selection we optimized for the best possible spatial distribution of events to improve coverage. The algorithm first picks a random event and then proceeds to choose that event from the whole remaining catalog that has the maximum distance to the next closest event. This guarantees the selection of all events with rare locations and approximates a Poisson disc sampling in regions with many events. Choosing more events would not add significant new information as each new event would be very close to an already existing one.

For each event we downloaded all freely accessible data from data centers worldwide resulting in recordings from about 2000 unique stations per event for legs B and C. The data sources are documented in section 5.6 and the total data coverage is illustrated in figure 5.6.

Before data and synthetics can be compared they must be filtered to enforce a similar spectral content, the influence of the recording instrument must be removed, and they must be sampled at the same points in time. In the interest of reproducibility, the processing steps and parameters are listed here:

1. Lowpass and decimate data close to the final desired sampling rate to speed up the following steps. The lowpass filter is a zero-phase type II Chebyschev filter.

2. Detrend by removing the best fitting linear function.

3. Taper the data to zero at both ends.

4. Instrument correct the data to velocity. Apply a frequency domain taper just outside the desired frequency band at the same time.

5. Sinc/Lanczos reconstruction/interpolation to the desired sampling rate. Note that we sample higher than required so the previously applied frequency domain taper ensures no aliasing.

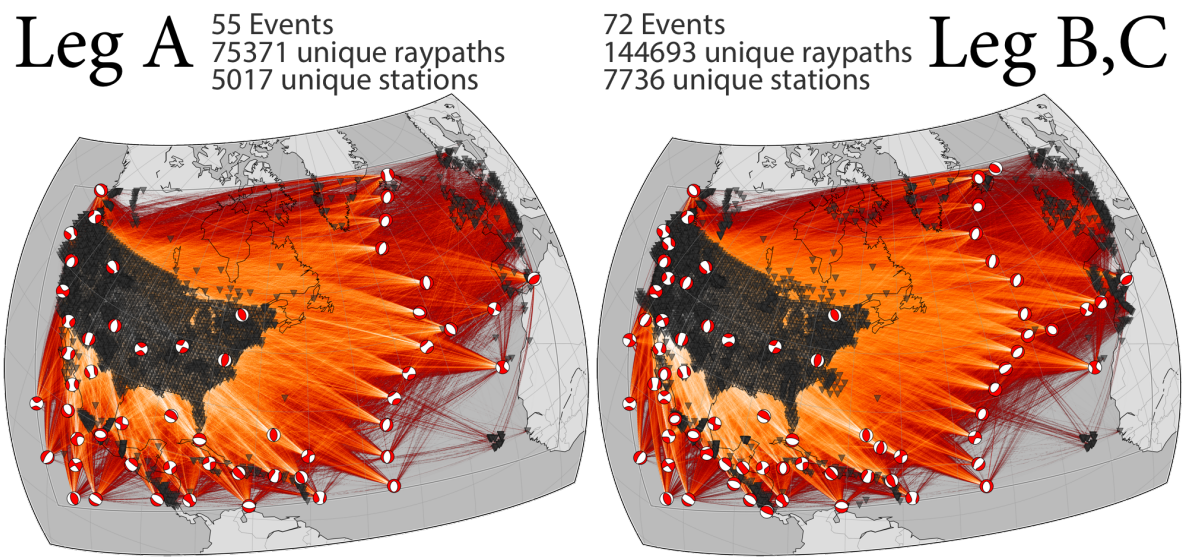

Figure 5.6.: Ray density plots of the used events and stations for the different legs of the inversion. Note that we added more (and removed some) events when going from leg A to B. Earthquakes are represented by red and white moment tensor beach-balls, stations by black triangles, and ray paths by red to white lines. The sensitivity of the measurements is, of course, not restricted to the ray paths but this is still a useful proxy to judge the potential resolution of the final model.

A similar frequency content for the synthetic data is asserted by convolving with a suitable source time function. The raw number of waveform traces prohibits manual control or visual inspection and thus has to be handled by automatic algorithms. Statistical plots like the one shown in figure 5.7 are used to judge their behaviour.

5.5 Results and Resolution Proxies

Any interpretation and further use of the derived model necessitates the knowledge of the uncertainties of each model parameter which can also be regarded as the reliability or trustworthiness of the model. Despite decades of theoretical and computational advances, it still remains a challenge. Linearizable inverse problems, given enough computational resources, allow the straightforward calculation of the resolution matrix and the posterior covariance of the solution. For larger problems this is not computationally tractable and only a subset of that information or some proxies can be retrieved. This section aims to enable readers to judge the quality and goodness of the model before presenting it.

5.5.1 Validation

A validation data set is used to help judge the quality of the final Earth model. This data set is made up of 10 earthquakes not used in the inversion procedure and is shown in figure 5.8. All in all it includes 22472 unique source-receiver raypaths recorded with mostly three components at 6166 unique stations. The data domain is slightly bigger than the inversion domain and contains some new stations, most notably in Spain and Morocco. The reason for this is that the validation data was acquired after the inversion has been completed and that data was not easily obtainable or discoverable at the time the inversion was performed. Slightly enlarging

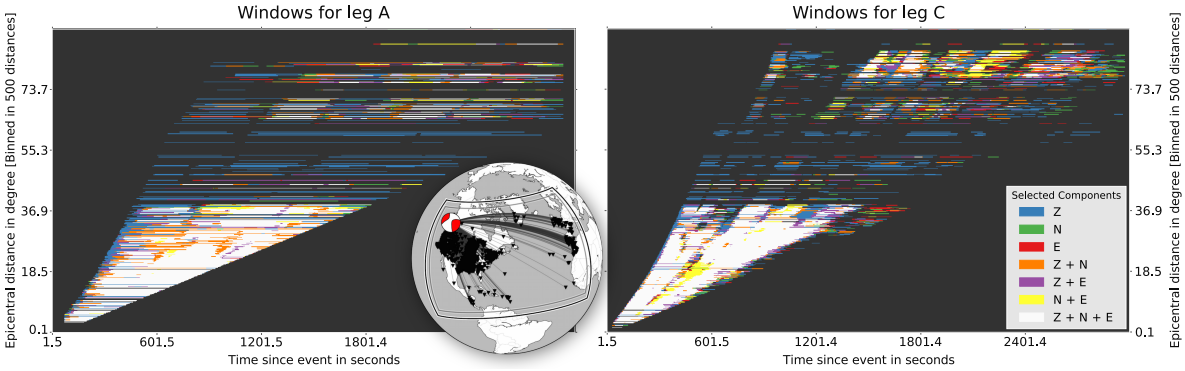

Figure 5.7.: Statistical visualization of the automatically picked windows for an $M_W = 6.5$ event close to Vancouver Island in September 2011. Once for leg A (left panel); once for leg C (right panel). The windows were picked at the beginning of each leg to not change the misfit definition within one L-BFGS run. Both panels show which combinations of channels were picked for each epicentral distance as a function of passed time since the event. The windows were automatically picked by LASIF. The inset globe shows the event and recording stations.

the domain was done to include a bit more, especially independent data compared to what was inverted for.

Synthetic seismograms with a duration of 3000 s are calculated through the initial and the final model with a mesh accurate for periods down to about 30 s. A different misfit calculation is required in order to give rise to a more independent validation. Inspired by Tape et al. (2010) and Simute et al. (2016) a normalized waveform difference L^2-misfit measurement is used to compare data and synthetics on all three components with no prior windowing or selection aside from some very basic signal to noise ratios to remove observed data traces that do not include any signal and thus would falsify the conclusion. χ_{wf} is given by

$$\chi_{wf}\left(\boldsymbol{m}\right) = \frac{\int_0^T \left[u_{obs}(t) - u_{syn}(t,\boldsymbol{m})\right]^2 dt}{\sqrt{\int_0^T \left[u_{obs}(t)\right]^2 dt \quad \int_0^T \left[u_{syn}(t,\boldsymbol{m})\right]^2 dt}}$$

where u_{obs} and u_{syn} are observed and corresponding synthetic waveform traces with data from time $t = 0$ to $t = T$ and $\boldsymbol{m}$ is the Earth model which the synthetic data depends on. Note that this measurement includes all parts of the seismograms, including very late arriving and scattered wave energy that was not inverted for and no amplitude normalization was applied.

75% of all seismograms improve when comparing against observed data going from the initial to the final model. Additionally, the summed misfit for each individual event decreases. Given that that the performed comparisons are the worst case scenario, this gives high confidence that the inversion yielded an improved Earth model. Figure 5.9 shows the waveform fit for events from the validation data set for the initial versus the final model.

5.5.2 Resolution Lengths

A more formal, quantitative resolution analysis is achieved via random probing after Fichtner and Leeuwen (2015). It yields position and direction dependent resolution lengths based on estimating the action of the Hessian on random test vectors. The resolution length is defined as the half-width of the point spread function in a certain direction and it can be interpreted as the minimum required distance to distinguish two separate perturbations. The main idea of the used method is that the action of the Hessian $H(\boldsymbol{m})$ on a perturbation $\delta\boldsymbol{m}$ can be interpreted

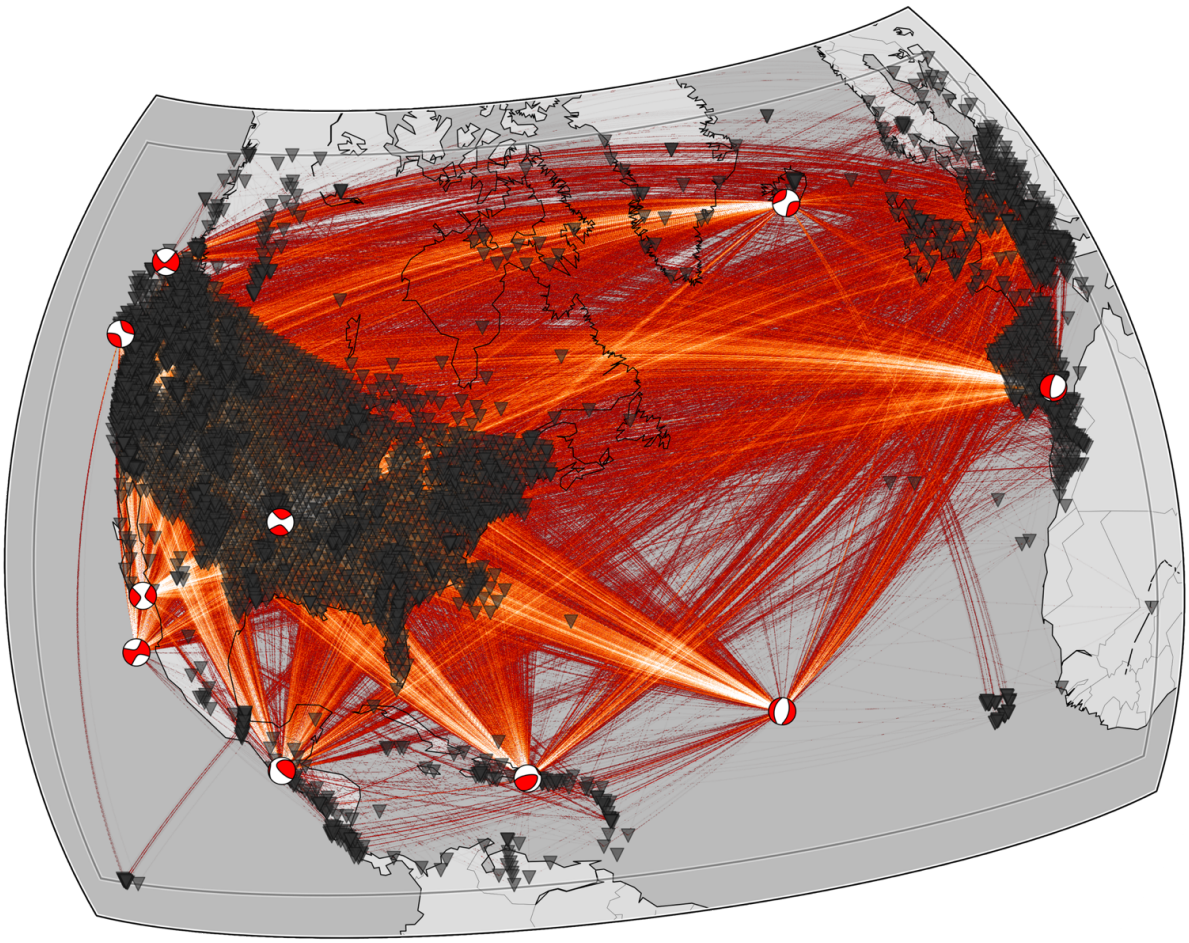

Figure 5.8.: Map with ten earthquakes that constitute the validation data set that were not used in the inversion. Event locations and focal mechanisms are shown as red-white beach balls, the recording stations are depicted as black triangles. The data consists of 22472 raypaths recorded on 6166 unique stations, mostly on all three components.

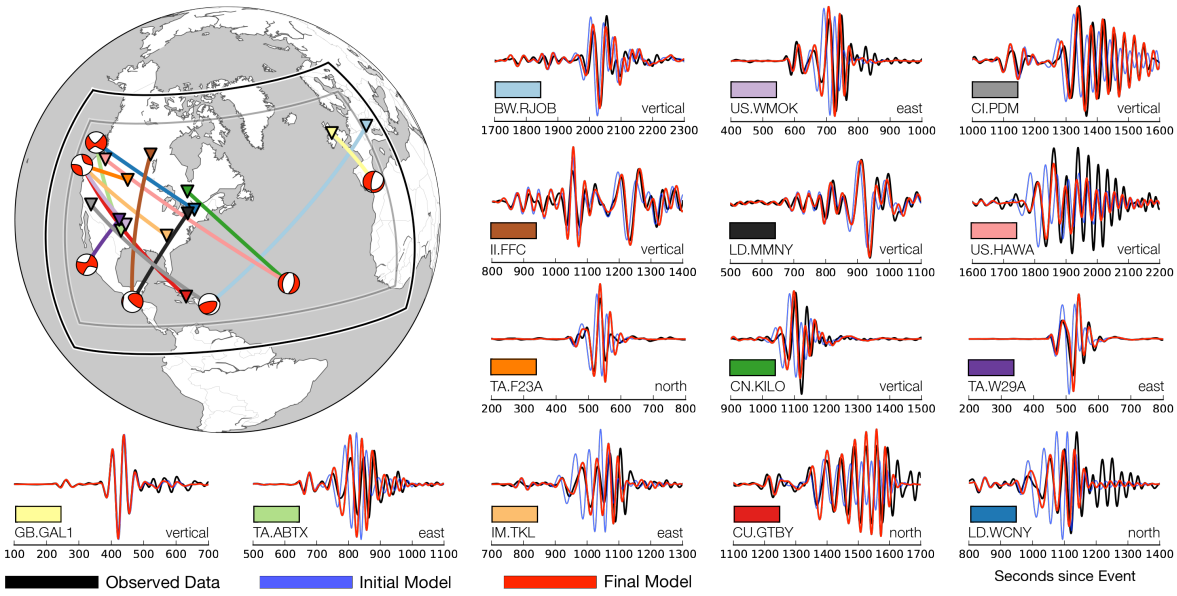

Figure 5.9.: Representative waveform plots for some events from the validation data set which was not inverted for comparing seismograms calculated through the initial model to seismograms calculated through the final model. Ray path colors match the boxes next to the seismograms. Note that the waveforms for some stations, for example for station GB.GAL1, basically did not change. This is due to the pre-existing full waveform inversion model for Europe.

as a conservative, worst case estimate of the point spread function. By applying random perturbations $\delta \boldsymbol{m}$ to a model and estimating $H(\boldsymbol{m})\delta \boldsymbol{m}$, its smoothing width in any direction can be estimated via auto correlations. Averaging over five random models yields stable resolution lengths estimates and the results are shown in figure 5.10.

It must be noted that $H(\boldsymbol{m})\delta \boldsymbol{m}$ was not calculated via true Hesse-vector products, which requires second-order adjoint simulations, but, for practical reasons, via gradient differences with first-order adjoints.

The strongly heterogeneous source-receiver distribution results in an uneven resolution of the resulting inverted model (see e.g. Rawlinson et al. (2014)). While the shown resolution lengths are calculated via simplifications and estimations and thus will contain artifacts, several trends can be observed. The resolution in the vertical direction is in general much better than in the horizontal directions. This is expected and a consequence of the surface wave dominated inversion and their depth sensitivity. The uneven ray-coverage in figure 5.6 foreshadows an anisotropic resolution in the Northern Atlantic region due to the predominantly E-W travelling waves which is confirmed in the resolution analysis.

An interesting thing to note is that model resolution is not equivalent to similarity to the true Earth but only gives a best-case scenario of the resolving power of the experimental and mathematical setup. The European part of the model is arguably one of the highest resolution parts of the model due to it originating from an existing high resolution full waveform inversion (see figure 5.2, Fichtner et al. (2013)). This chapter's inversion only has comparatively little data for that region and its resolution analysis consequently claims no resolved fine-scale features.

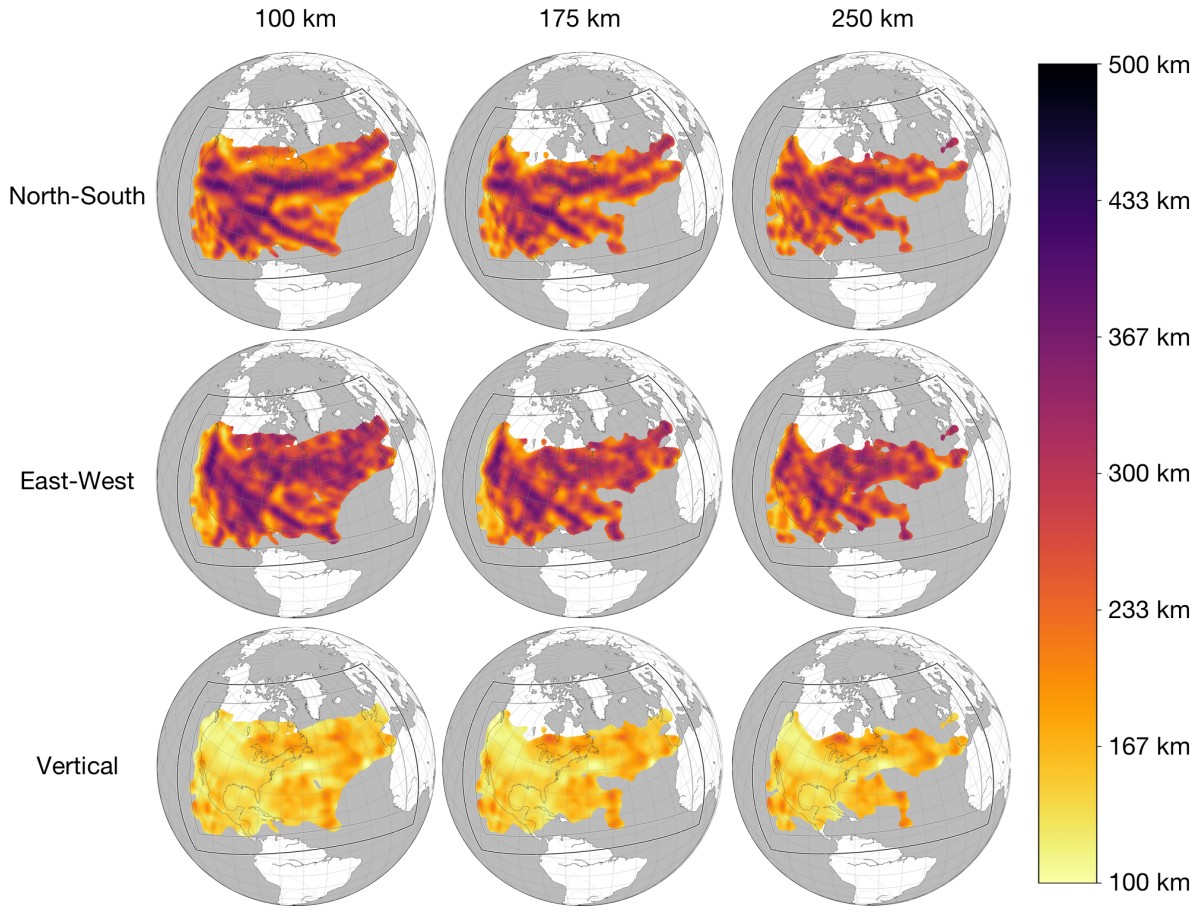

Figure 5.10.: Position and direction dependent resolution lengths based on estimating the action of the Hessian on a random test vector calculated with the random probing technique of Fichtner and Leeuwen (2015) for the final model. The top row shows resolution lengths approximately in North-South direction along the mesh boundaries, the center row in East-West direction, and the last row demonstrates vertical resolution lengths. The columns are depth slices in 100 km, 175 km, and 250 km depth. No data is shown in regions where the absolute sum of all considered kernels to estimate the Hessian is beneath a chosen threshold as these could not be interpreted. Generally visible trends include the much better vertical compared to horizontal resolution due to the dominant usage of surface waves in the inversion, better N-S resolution in large parts of the Northern Atlantic compared to the E-W resolution as expected from the dominantly E-W travelling waves in that region (compare to figure 5.6), and a shrinking region of good data coverage with depth.

5.5.3 Neglected Sources of Errors

The preceding paragraphs discuss the errors within our chosen mathematical framework. It is important to understand that these results have to be interpreted as a best case scenario, e.g. the true parameter errors and resolution lengths cannot be better and are likely worse as there are a number of neglected error sources in that analysis. These are discussed here.

While we do employ a physically accurate numerical scheme to calculate forward and adjoint wave fields, which in theory yields accurate gradients, there are still a number of physical effects and facts we do not account for. Some can be safely neglected within our chosen period range, others should be accommodated for in future studies.

Komatitsch and Tromp (2002b) demonstrate how to add the physics of oceans, rotation, and self-gravitation to spectral element simulations - we simulate neither of them, mainly for reasons of code complexity, practicality, and computational efficiency. Rotation has a very small potential effect on the amplitudes which is of no further concern for the study at hand as it uses a phase-only time-frequency misfit. Self-gravitation has the most profound effect on long period waves and its influence is only minor for the frequencies we invert for. Including even lower frequencies would indeed necessitate self-gravitating waves. However, ocean loading can have a measurable effect, particularly on Rayleigh waves, and this should thus be taken into consideration in future studies.

The used numerical mesh has some inaccuracies as well. It is a regular, spherical chunk and thus has no topography, ellipticity, nor internal boundaries. Topography and ellipticity have only minor influences for surface waves at long periods. Nuber et al. (2016) show the influence of topography effects on full waveform inversions. They conclude that it does affect the final result but only for significant topography, which our chosen inversion domain does not have compared to its total scale and inverted wavelengths. The most significant internal boundary in our inversion domain is the Moho. It is not honored by our mesh so the final inverted model has to be interpreted as an effective representation of the true crustal model for our mesh and frequency range (Backus, 1962; Capdeville, 2010; Fichtner and Igel, 2008). Future studies could directly mesh the internal discontinuities but this brings along a host of new problems if they are not exactly known. If the boundaries are not also inverted for, the inversion procedure will try to find the best fitting model given these pre-imposed discontinuities which might differ from the true ones. The consequences of this should be studied in more detail.

Further errors we do no take into account are errors in the observed data. These can and likely do range from faulty sensors, inaccurate orientations, very strong and local site effects, timing errors, wrong instrument responses, and a whole host of other problems. This, as a whole, is a problem that is almost intractable as many subtle errors can only be figured out by the network and data center operators, potentially with the help of projects like IRIS' MUSTANG (http://service.iris.edu/mustang). Our quality control, as described in the data section, catches many of these by comparing observed to synthetic data but subtly erroneous data will still enter the inversion. Using a lot of data should reduce the effects of misbehaving seismograms. Properly quantifying and propagating data errors will remain to be a challenge for the foreseeable future.

Last but not least, we also do not invert for the sources but assume the earthquakes from the GCMT catalog (Ekström et al., 2012) to be correct. Our inversion domain is not global so any inversion for source mechanisms will likely incur a bias from imperfect azimuthal coverage especially for those events close to the domain boundary. This brings along a risk of overfitting to the data at hand. Bozdağ et al. (2016) perform a global full waveform source inversion for 253

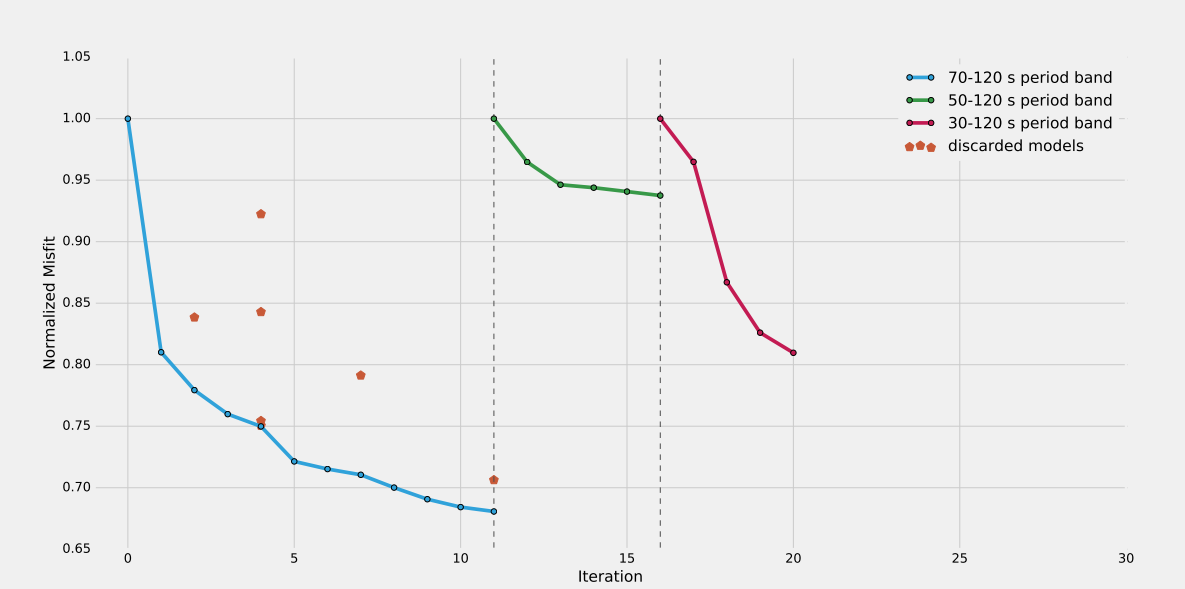

Figure 5.11.: Misfit evolution and failed test models over the course of the inversion's three legs. The orange-red pentagons denote discarded test models where the step-length was too large. The blue line denotes the misfit decrease during the 70-120 s leg A of the inversion, the green line the reduction during leg B with periods from 50-120 s, and the red line finally showing the misfit evolution during the computationally expensive leg C of the operation from 30-120 s.

events from the GCMT catalog and find that they generally change very little, usually less than 5 km. This is consistent with findings in Hjörleifsdóttir and Ekström (2010) which presents the results of synthetic tests for the accuracy of the GCMT catalog. Concluding, it might benefit the inversion to invert for some source parameters, most prominently the event depth, but the risk of a bias especially in restricted domains has to be accounted for. The frequency range of this manuscript's inversion and the expected differences in event parameters would only result in a minor effect.

5.5.4 Final Model

After discussing the robustness, resolution power and potential artifacts of the model this section describes the evolution of the model and especially its final state. All in all we carried out 20 iterations over three legs in the course of the inversion as shown in figure 5.11, each leg a single L-BFGS optimization run. 11 iterations were performed for the longest period leg A at 70-120 s. For one, this was computationally comparatively cheap and the applied strong smoothing of the gradients also ensures to the largest part that the model does not get trapped in a local minimum. Further iterations at 70 s would not have significantly improved the overall misfit. Even though we started with a high-quality 3-D model we managed to significantly reduce the misfit in this leg.

For leg B, at a period range of 50-120 s, we only performed 5 iterations as the misfit evolution stagnated as seen in the figure. A potential cause for this is that there is not that big a difference between simulations at 70 s and at 50 s and the model at 70 s already largely converged. Furthermore, the initial model of course also had contributions at smaller periods. Leg A improved the long period parts of it and leg B then only had to invert for a small period range.

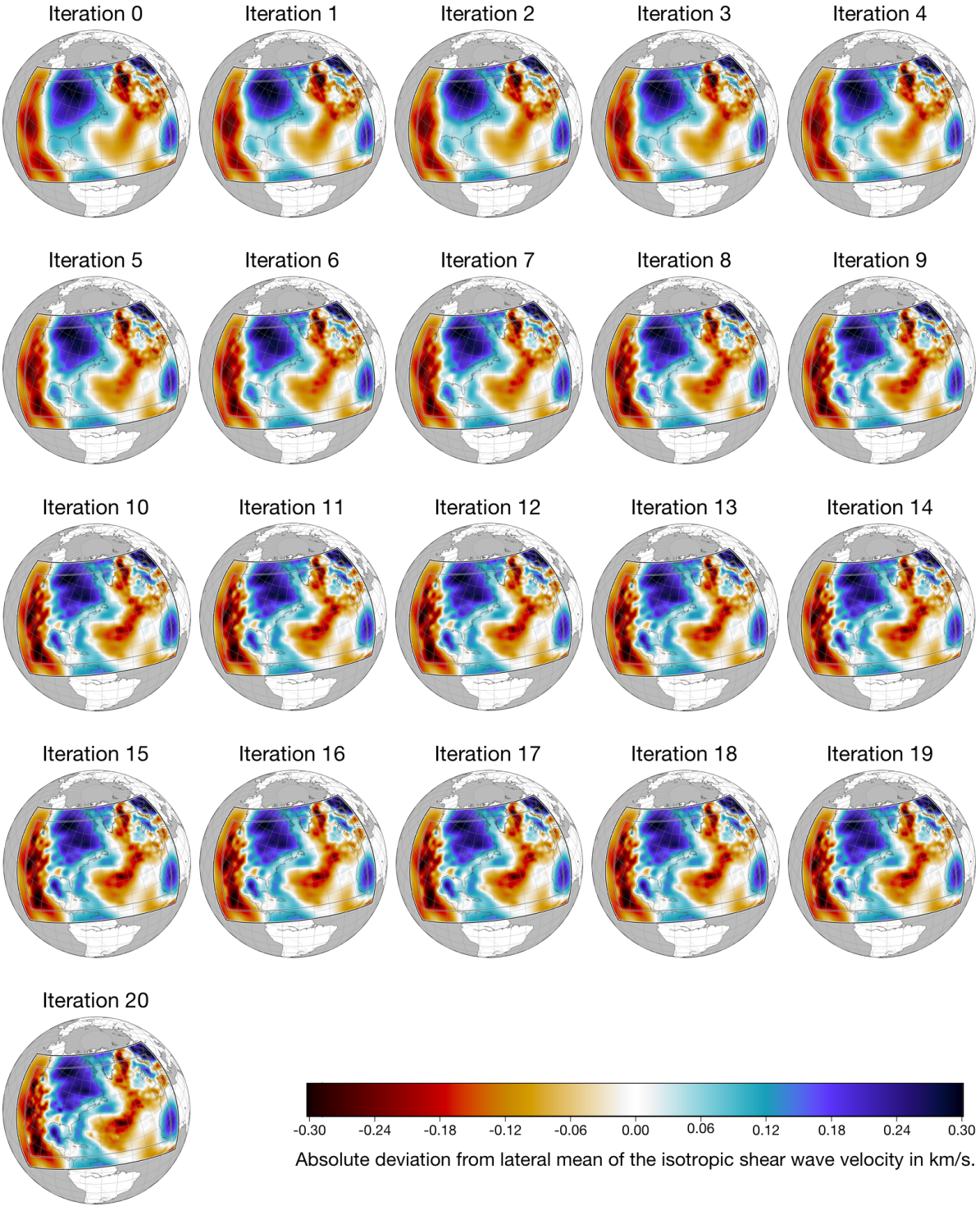

Figure 5.12.: Horizontal slices of the model and its evolution at every iteration. It shows absolute deviations of the isotropic shear wave velocity from the lateral mean at a depth of 150 km at a frequency of 1 Hz. It demonstrates the need for the used multi-scale approach for the chosen setup as the inversion, as desired, first fixes the large-scale structures before successively adding finer details.

Simulations become significantly more expensive at periods from 30-120 s but do result in a fast and strong misfit reduction which only lasts for a single iteration before tapering off. We performed a couple more iterations after iteration 20 which still decrease the total misfit but strong localized effects, mainly around the sources are starting to appear so we terminated the inversion at this point in order to avoid over-fitting and an unphysical model. In future works, we plan to evaluate different data regularization and weighting schemes attempting to further push the resolution. A careful inversion for source parameters at this stage might also help but would not avoid the risk of over-fitting as detailed in the previous section. Breaking the model into multiple smaller domains would also allow pushing to higher resolutions as the then usable smaller events would improve the data coverage.

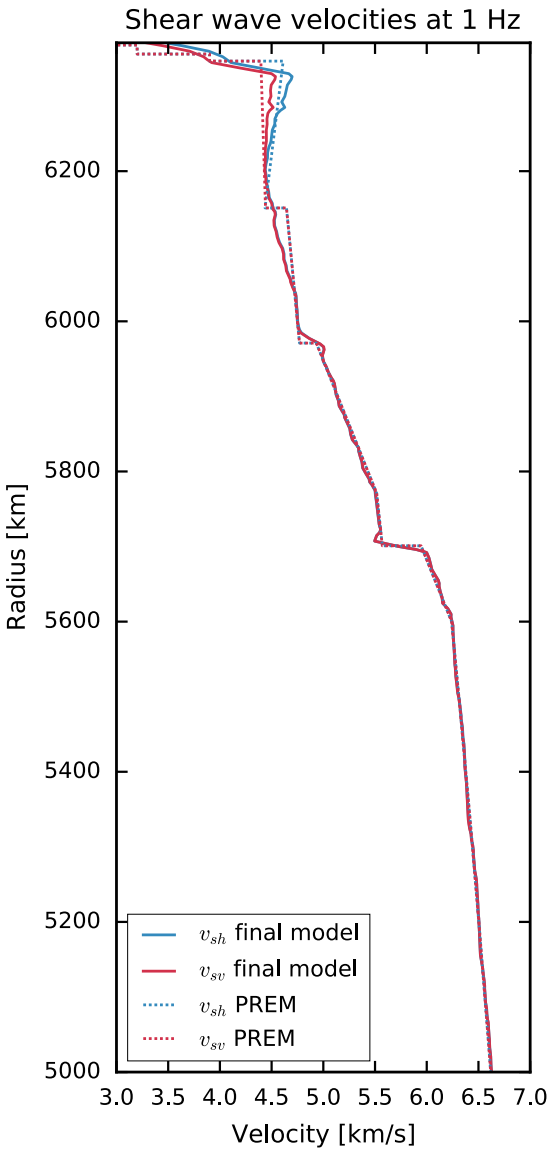

Figure 5.13.: Lateral averages of the horizontally and vertically polarized shear wave velocities of the final model compared to PREM (Dziewonski and Anderson, 1981) at 1 Hz. The 220 km discontinuity has been smoothed out in the initial model, the inversion replaced the others with effective versions.

Note the orange-red pentagons in figure 5.11 representing rejected test models that failed one of the Wolfe conditions written about in section 5.2.2. L-BFGS results in a natural step length of one, under the assumption that the misfit is locally quadratic. For our non-linear problem that is not always the case and this manifests in these discarded test models. A reduction of the step-length was required in these cases to further reduce the misfit. By and large the L-BFGS algorithm works really well in the large-scale example studied in this chapter and mostly eliminates the need to perform an explicit and costly line-search to determine a suitable step length. The simulations to test a misfit value immediately become the forward simulations of the next iteration greatly speeding up the process. The internal smoothing in the L-BFGS parameterization successfully manages to avoid the need to restart the optimization algorithm that has been observed in other studies.

Figure 5.12 presents the model evolution with horizontal slices of isotropic shear wave speed at a depth of 150 kilometers for all iterations. As expected, the first iterations only change the large-scale structure while later iterations keep adding more details. Key insights from that figure are that the gradient based optimization scheme improves the models only slowly with fairly small differences from iteration to iteration. Large jumps in misfit value also do not coincide with big apparent changes in the model. It also makes a strong case for the chosen

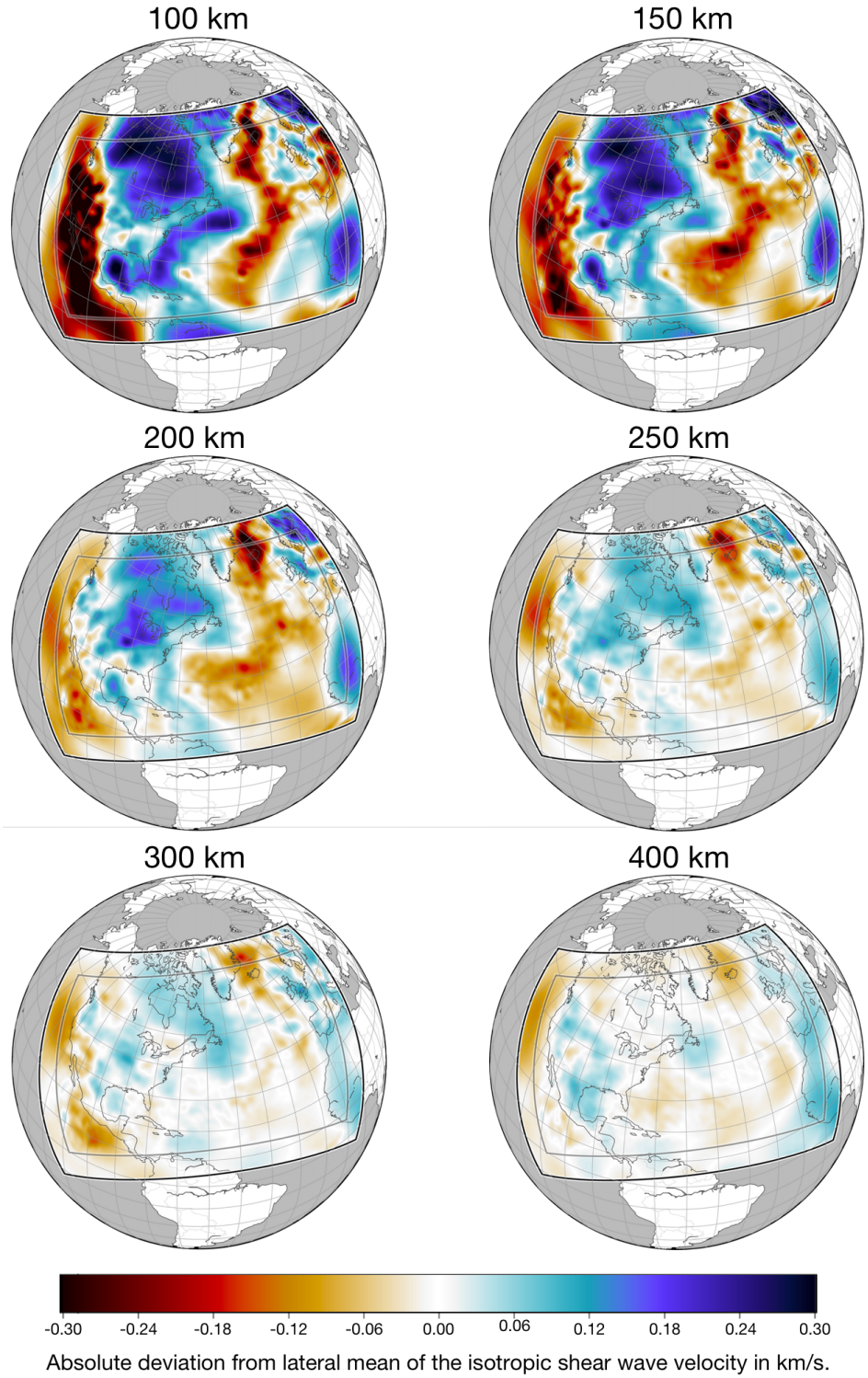

Figure 5.14.: Horizontal slices through the final model showing the isotropic shear wave speed at various depths with the same color scale. The absolute (and also the relative) deviations from the lateral mean decrease with increasing depth. Very shallow layers of the crust are not shown as they cannot really be resolved with the chosen period bands and show effective rather than true elastic parameters.

multi-scale inversion strategy. Directly inverting for smaller period data might have resulted in a different model representing a different and likely worse local minimum. The computational costs for such an inversion are unfortunately too high to test that hypothesis.

An encouraging result is that the final model in Europe does not significantly differ from the initial model. The final model continues pre-existing detailed features into the low resolution region of the initial model which is best visible along the Mid-Atlantic Ridge. As previously mentioned, the initial model's European part originated from another full waveform inversion operating in a similar period band (Fichtner et al., 2013) but with a very different data set and partially different methodology. We did not treat the European region of the model any different to the rest of the domain. The same is true for the part of the model including results from an inversion for the South Atlantic region (Colli et al., 2013) but, as we have no data there, that is hardly surprising.

The lateral v_{sh} and v_{sv} averages of the inversion domain against their PREM (Dziewonski and Anderson, 1981) counterparts are plotted in figure 5.13. Lessons to draw from this figure are, that the final model mostly has sensitivity to the upper 300 kilometers, which is in accordance with it inverting mostly for fairly long period surface waves. We also included body waves in the inversion where they could be measured but their influence on the final model is limited: For one long distance body waves travel fairly deep and they were largely absorbed by the bottom domain boundary. Additionally, we did not treat them any different than the surface waves. As even a radially symmetric Earth model can explain body wave travel times to a remarkable degree, the expected phase differences are very small as is their influence on the resulting gradients and thus the final model. More sophisticated weighting schemes would be a way to resolve this but in this study we focused on using large and continuous data windows without worrying which phases they contain.

Figure 5.14 shows a different perspective on this by plotting horizontal slices in depth. As in figure 5.13, it shows the heterogeneities with respect to the lateral mean to decrease with depth in absolute as well as relative terms. The crust cannot truly be resolved with the frequencies we are inverting for in this study. The final model's crust still has a strong impact from the initial model and the inversion to some extent modified it to act as an effective crust for the frequency range we inverted for. Strong contrasts are still largely preserved from the initial model as smooth model updates will not significantly change these. Thus the crust is to some extent a version that works for this particular inversion but cannot be interpreted in a geophysical manner and we thus choose to not show it here.

A future study in collaboration with tectonophysicists, geologists, and geodynamisists is required to carefully analyze the features of the model taking into consideration its uncertainties and augmenting and synthesizing it with other types of data and theoretical arguments. Here, we only point out some of the model's features.

The final version contains a strong imprint of the Mid-Atlantic-Ridge. It is not continuous on the southern edge of the model but that is likely an artifact of lacking resolution in that region. Other traits of the model are the remarkably homogeneous Canadian shield, the visible Mississippi delta and Yellowstone volcano, as well as the low resolution zone beneath the Golf of Mexico. Also note the depth extent of the Island-Jan Mayen plume system which has the greatest depth extent of any feature in the model. It has been studied in more detail in Rickers et al. (2013).

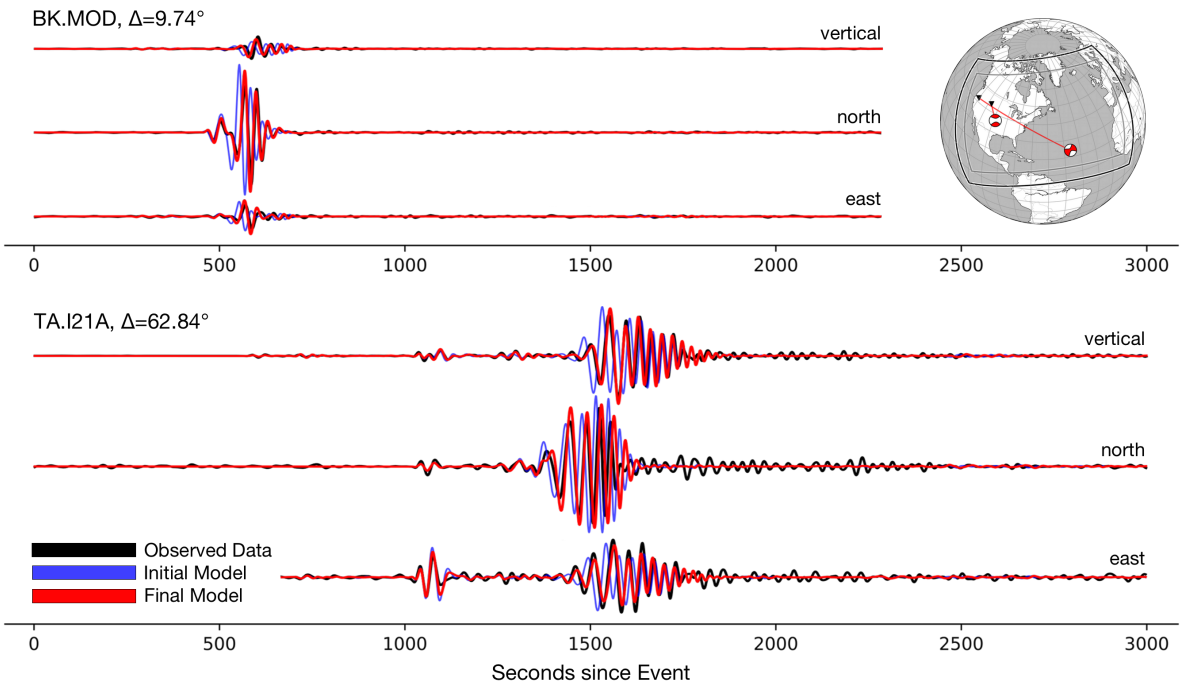

Figure 5.15.: Exemplary waveform comparisons of synthetics through the initial and the final model simulated at a dominant period of 30 s against observed data. Amplitudes are not scaled. The inversion used windowed data but these two examples show waveform traces for the full simulated duration. Surface waves are affected more strongly than body waves whose effect cannot be seen at this scale. Phases are greatly improved throughout the whole data set, amplitudes are, in general, not affected a lot.

While the strength of heterogeneities in general decreases with depth our model enhances that effect by having less sensitivity in deeper regions. That has to be taken into account when interpreting the model.

The seismic velocity model is the most important result of this study. It produces synthetic seismograms that are objectively more similar to the observed data than seismograms calculated with the initial model. Figure 5.15 shows two exemplary three component recordings against synthetics calculated with the initial model and the final model for the full duration of the simulations. Note that we do not only fit a few phases but basically complete seismograms including most complexities. We did not invert for amplitudes and the examples here are to some extent picked and chosen to also fit the amplitudes but the very good phase fit of data and synthetics can be observed throughout the data set.

The waveform plots make up yet another argument for multi-scale inversions: Some wiggles of the seismograms from the initial model are cycle skipped and a meaningful phase misfit measurement would not have been possible and would have resulted in a wrong local minimum.

5.6 Conclusion

This chapter presented a new full seismic waveform inversion model for North America and the Northern Atlantic using USArray and other data. The model was constructed using numerical forward and adjoint simulations with the spectral element method coupled with a time-frequency domain phase misfit. Updates for 20 iterations were carried out with the L-BFGS method. To judge the quality of the final model we performed a validation test with

new data and an independent misfit measurement and the final model improved the fit to the data also for that data set. Additionally, we performed a quantitative resolution analysis and debated other potential sources of errors before discussing the final model and its features.

Future works need to find a way to further push the resolution without running into strongly local artifacts. Using even more data might help with that but further problems likely have to be alleviated beforehand. This for one includes more physically valid simulations by including in particular the effect of the ocean mass on the seismic wave field. Additional improvements could be gained by adding internal and external boundary topography, ellipticity and the effect of Earth's rotation. Another promising route would be to use more sophisticated data weighting schemes, regularization techniques, and potentially pre-conditioners to handle the spatially strongly varying data availability. Weighting body wave measurements in a different manner would also allow for more resolution in greater depths. Exploiting the strong spatial clustering of seismic sources as well as receivers via special misfits and adjoint sources also has great potential to improve the resolution (Yuan et al., 2016).

Acknowledgements

Computations were performed on the Leibniz Supercomputing Center (LRZ), whose support is gratefully acknowledged. Event data was acquired from the Global CMT Catalog (Ekström et al., 2012) as written in section 5.4. Nothing in this chapter would have been possible without the data centers and network operators generously offering their services and data. That support is recognized here. We acquired freely available data from data centers listed in the following together with the URLs (all last accessed March 2017) of their FDSN (Romanowicz and Dziewonski, 1986) compatible web services:

- Bundesanstalt für Geowissenschaften und Rohstoffe (BGR, Federal Institute for Geosciences and Natural Resources) in Hannover: `http://eida.bgr.de`
- Swiss Seismological Service (SED) at ETH Zurich: `http://eida.ethz.ch`
- Helmholtz Centre Potsdam - GFZ German Research Centre for Geosciences: `https://geofon.gfz-potsdam.de`
- Istituto nazionale di geofisica e vulcanologia (INGV): `http://webservices.rm.ingv.it`
- Institut de physique du globe de Paris (IPGP): `http://eida.ipgp.fr`
- Incorporated Research Institutions for Seismology Data Management Center (IRIS DMC): `http://service.iris.edu`
- Kandilli Observatory And Earthquake Research Institute (KOERI): `http://eida.koeri.boun.edu.tr`
- Ludwig-Maximilians-University Munich (LMU): `http://erde.geophysik.uni-muenchen.de`
- Northern California Earthquake Data Center (NCEDC): `http://service.ncedc.org`
- Observatories & Research Facilities for European Seismology (ORFEUS): `http://www.orfeus-eu.org`
- Réseau Sismologique et Géodesique Français (RESIF): `http://ws.resif.fr`

- Southern California Earthquake Data Center (SCEDC): `https://service.scedc.caltech.edu`

From these data centers we obtained waveforms and station meta-information from 140 permanent networks and 104 temporary deployments. Digital object identifiers (DOIs) are starting to be used to link to data from seismic networks in order to give credit where it's due (Evans et al., 2015). As best practices in this regard are not yet fully established and moving everything to the bibliography is impractical for data-heavy projects we are listing them here in alphabetical order, starting with the permanent networks:

- **AE**: Arizona Geological Survey (2009): Arizona Broadband Seismic Network. doi:10.7914/SN/AE
- **AF**: Penn State University (2004): AfricaArray. doi:10.7914/SN/AF
- **AK**: Alaska Earthquake Center, Univ. of Alaska Fairbanks (1987): Alaska Regional Network. doi:10.7914/SN/AK
- **AT**: NOAA National Oceanic and Atmospheric Administration (USA) (1967): National Tsunami Warning Center Alaska Seismic Network. doi:10.7914/SN/AT
- **AZ**: Frank Vernon, UC San Diego (1982): ANZA Regional Network. doi:10.7914/SN/AZ
- **BC**: Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Ensenada (1980): Red Sismica del Noroeste de Mexico. doi:10.7914/SN/BC
- **BE**: Royal Observatory of Belgium (1985): Belgian Seismic Network. doi:10.7914/SN/BE
- **BK**: Northern California Earthquake Data Center. (2014). Berkeley Digital Seismic Network (BDSN) [Data set]. Northern California Earthquake Data Center. doi:10.7932/BDSN
- **CA**: Institut Cartogràfic i Geològic de Catalunya-Institut d'Estudis Catalans (1996): Catalan Seismic Network. doi:10.7914/SN/CA
- **CC**: Cascades Volcano Observatory/USGS (2004): Cascade Chain Volcano Monitoring. doi:10.7914/SN/CC
- **CH**: Swiss Seismological Service (SED) at ETH Zurich. (1983). National Seismic Networks of Switzerland. ETH Zürich. doi:10.12686/sed/networks/ch
- **CI**: California Institute of Technology (Caltech) (1926): Southern California Seismic Network. doi:10.7914/SN/CI
- **CO**: University of South Carolina (1987): South Carolina Seismic Network. doi:10.7914/SN/CO
- **CU**: Albuquerque Seismological Laboratory (ASL)/USGS (2006): Caribbean USGS Network. doi:10.7914/SN/CU
- **CZ**: Institute of Geophysics, Academy of Sciences of the Czech Republic (1973): Czech Regional Seismic Network. doi:10.7914/SN/CZ
- **DR**: Universidad Autonoma de Santo Domingo (ISU/UASD Dominican Republic) (1998): Centro Nacional de Sismologia (CNS). doi:10.7914/SN/DR
- **EI**: Dublin Institute for Advvanced Studies (1993): Irish National Seismic Network (INSN). doi:10.7914/SN/EI
- **FR**: RESIF. (1995). RESIF-RLBP French Broad-band network, RESIF-RAP strong motion network and other seismic stations in metropolitan France. RESIF - Réseau Sismologique et géodésique Français. doi:10.15778/RESIF.FR
- **G**: Institut de Physique du Globe de Paris (IPGP), & Ecole et Observatoire des Sciences de la Terre de Strasbourg (EOST). (1982). GEOSCOPE, French Global Network of broad band seismic stations. Institut de Physique du Globe de Paris (IPGP). doi:10.18715/GEOSCOPE.G
- **GE**: GEOFON Data Centre. (1993). GEOFON Seismic Network. Deutsches GeoForschungsZentrum GFZ. doi:10.14470/TR560404
- **GS**: Albuquerque Seismological Laboratory (ASL)/USGS (1980): US Geological Survey Networks. doi:10.7914/SN/GS
- **GU**: University of Genova (1967): Regional Seismic Network of North Western Italy. doi:10.7914/SN/GU
- **IB**: Institute Earth Sciences "Jaume Almera" CSIC (ICTJA Spain) (2007): IberArray. doi:10.7914/SN/IB
- **IE**: Idaho National Laboratory (1972): INL Seismic Monitoring Program. doi:10.7914/SN/IE
- **II**: Scripps Institution of Oceanography (1986): IRIS/IDA Seismic Network. doi:10.7914/SN/II
- **IU**: Albuquerque Seismological Laboratory (ASL)/USGS (1988): Global Seismograph Network (GSN - IRIS/USGS). doi:10.7914/SN/IU
- **IV**: INGV Seismological Data Centre. (1997). Rete Sismica Nazionale (RSN). Istituto Nazionale di Geofisica e Vulcanologia (INGV), Italy. doi:10.13127/SD/X0FXnH7QfY
- **IW**: Albuquerque Seismological Laboratory (ASL)/USGS (2003): Intermountain West Seismic Network. doi:10.7914/SN/IW
- **KP**: Korea Polar Research Institute (KOPRI) (2013): Korea Polar Seismic Network. doi:10.7914/SN/KP
- **KY**: Kentucky Geological Survey/Univ. of Kentucky (1982): Kentucky Seismic and Strong Motion Network. University of Kentucky. doi:10.7914/SN/KY
- **LI**: California Institute of Technology (Caltech) (2000): Laser Interferometer Gravitational-Wave Experiment (LIGO). doi:10.7914/SN/LI
- **LO**: Instituto Politecnico Loyola (2012): Observatorio Sismológico Politécnico Loyola. doi:10.7914/SN/LO
- **MB**: Montana Bureau of Mines and Geology/Montana Tech (MBMG, MT USA) (2001): Montana Regional Seismic Network. doi:10.7914/SN/MB
- **MN**: MedNet project partner institutions. (1988). Mediterranean Very Broadband Seismographic Network (MedNet). Istituto Nazionale di Geofisica e Vulcanologia (INGV), Italy. doi:10.13127/SD/fBBBtDtd6q
- **N4**: UC San Diego (2013): Central and Eastern US Network. doi:10.7914/SN/N4
- **NC**: USGS Menlo Park (1967): USGS Northern California Network. doi:10.7914/SN/NC
- **NE**: Albuquerque Seismological Laboratory (ASL)/USGS (1994): New England Seismic Network. doi:10.7914/SN/NE
- **NI**: OGS (Istituto Nazionale di Oceanografia e di Geofisica Sperimentale) and University of Trieste (2002): North-East Italy Broadband Network. doi:10.7914/SN/NI
- **NN**: University of Nevada (UNR Reno) (1971): Nevada Seismic Network. doi:10.7914/SN/NN
- **NP**: USGS Earthquake Science Center (1931): United States National Strong-Motion Network. doi:10.7914/SN/NP
- **NU**: Nicaraguan Institute of Terristerial Studies (1992): Nicaraguan Seismic Network. doi:10.7914/SN/NU
- **NX**: Nanometrics Seismological Instruments (2013): Nanometrics Research Network. doi:10.7914/SN/NX
- **NY**: Pascal Audet, University of Ottawa (2013): Yukon-Northwest Seismic Network. doi:10.7914/SN/NY
- **OK**: Oklahoma Geological Survey (1978): Oklahoma Seismic Network. doi:10.7914/SN/OK

- **OO**: Rutgers University (2013): Ocean Observatories Initiative. doi:10.7914/SN/OO
- **OX**: OGS (Istituto Nazionale di Oceanografia e di Geofisica Sperimentale) (2016): North-East Italy Seismic Network. doi:10.7914/SN/OX
- **PE**: Penn State University (2004): Pennsylvania State Seismic Network. doi:10.7914/SN/PE
- **PR**: University of Puerto Rico (1986): Puerto Rico Seismic Network (PRSN) & Puerto Rico Strong Motion Program (PRSMP). doi:10.7914/SN/PR
- **PY**: Frank Vernon, UC San Diego (2014): Piñon Flats Observatory (PFO) Array. doi:10.7914/SN/PY
- **RC**: Brigham Young Univ-Idaho (BYU Idaho) (2001): BYU-Idaho Network. doi:10.7914/SN/RC
- **RV**: Alberta Geological Survey / Alberta Energy Regulator (2013): Regional Alberta Observatory for Earthquake Studies Network. doi:10.7914/SN/RV
- **SL**: Slovenian Environment Agency (2001): Seismic Network of the Republic of Slovenia . doi:10.7914/SN/SL
- **SN**: University of Nevada (UNR Reno) (1995): Southern Great Basin Network. doi:10.7914/SN/SN
- **ST**: Geological Survey-Provincia Autonoma di Trento (1981): Trentino Seismic Network. doi:10.7914/SN/ST
- **TA**: IRIS Transportable Array (2003): USArray Transportable Array. doi:10.7914/SN/TA
- **TC**: InformaciÃ³n de la Red SismolÃ³gica Nacional de Costa Rica. (n.d.). doi:10.15517/tc
- **TE**: Electricity Generating Authority of Thailand (2015): EGAT Dams Seismic Monitoring System. doi:10.7914/SN/TE
- **UP**: SNSN. (1904). Swedish National Seismic Network. Uppsala University, Uppsala, Sweden. doi:10.18159/SNSN
- **US**: Albuquerque Seismological Laboratory (ASL)/USGS (1990): United States National Seismic Network. doi:10.7914/SN/US
- **UU**: University of Utah (1962): University of Utah Regional Seismic Network. doi:10.7914/SN/UU
- **UW**: University of Washington (1963): Pacific Northwest Seismic Network. doi:10.7914/SN/UW
- **WI**: Institut de Physique du Globe de Paris- IPGP. (2008). GNSS, seismic broadband and strong motion permanent networks in West Indies. Institut de Physique du Globe de Paris - IPGP. doi:10.18715/antilles.WI
- **WM**: San Fernando Royal Naval Observatory (ROA), Universidad Complutense de Madrid (UCM), Helmholtz-Zentrum Potsdam Deutsches Geoforschungszentrum (GFZ), Universidade de Evora (UEVORA, P.), & Institute Scientifique of RABAT ISRABAT, M. (1996). The Western Mediterranean BB seismic Network. Deutsches Geoforschungszentrum GFZ. doi:10.14470/JZ581150
- **WY**: University of Utah (1984): Yellowstone Wyoming Seismic Network. doi:10.7914/SN/WY

In addition, we used data from the following permanent seismic networks that did not yet have a DOI assigned at the time of writing: AG, AO, AR, AX, AY, BN, BW, CM, CN, CW, CY, DK, DZ, EB, EE, EP, ES, ET, GB, GI, GL, GR, HE, HF, HW, IM, IP, JM, LB, LC, LD, LX, MC, MG, MQ, MT, MX, NA, NL, NM, NO, NR, NS, NV, NW, OE, OV, PA, PB, PG, PL, PM, PN, PO, RD, RE, S, SB, SC, SE, SI, SS, SV, SX, TD, TH, TO, TR, TV, UK, UO, VE, VI, WC, WR, and WU.

Used temporary networks are postfixed with an underscore and the year of the deployment, again in alphabetical order:

- **2G_2010**: Bruce Douglas, Gary Pavlis, Jon Cameron (2010): Testing of the effectiveness of incorporating seismic data in seismic hazard assessment within a traditional field course. doi:10.7914/SN/2G_2010
- **3D_2010**: Christine Thomas (2010): Morocco-Muenster. doi:10.7914/SN/3D_2010
- **3E_2010**: Jer-Ming Chiu (2010): Exploring Seismic Velocity of Sediments in the Mississippi Embayment. doi:10.7914/SN/3E_2010
- **4F_2015**: Heather DeShon, Chris Hayward, Jacob Walter, M. Beatrice Magnani, Matthew Hornbach, Brian Stump (2015): North Texas Earthquake Study: Venus (Johnson County), TX. doi:10.7914/SN/4F_2015
- **5A_2010**: John Nabelek (2010): Crustal-Scale Geometry of Active Continental Normal Faults. doi:10.7914/SN/5A_2010
- **5E_2014**: Anne Trehu (2014): Cascadia Initiative. doi:10.7914/SN/5E_2014
- **7A_2013**: Maureen Long, Paul Wiita (2013): Mid-Atlantic Geophysical Integrative Collaboration. doi:10.7914/SN/7A_2013
- **7D_2009**: Wes Thelen, Paul Bodin (2009): Yakima Landslide RAMP. doi:10.7914/SN/7D_2009
- **7D_2011**: IRIS OBSIP (2011): Cascadia Initiative Community Experiment - OBS Component. doi:10.7914/SN/7D_2011
- **7E_2006**: Wilde-Piórko, M., Geissler, W. H., Plomerová, J., Knapmeyer-Endrun, B., Grad, M., Babuška, V., … Wiejacz, P. (2006). PASSEQ 2006-2008: Passive Seismic Experiment in Trans-European Suture Zone. Deutsches Geoforschungszentrum GFZ. https:// doi:10.14470/2R383989
- **9D_2010**: Roman Motyka, Mark Fahnestock, Martin Truffer (2010): Ice-ocean interaction at Nuuk tidewater glaciers. doi:10.7914/SN/9D_2010
- **X4_2016**: Recep Cakir (2016): Active Fault Mapping. doi:10.7914/SN/X4_2016
- **X8_2012**: William Menke, Vadim Levin, Fiona Darbyshire (2012): Deep Structure of Three Continental Sutures in Eastern North America. doi:10.7914/SN/X8_2012
- **X9_2008**: Brian Stump, Chris Hayward (2008): Dallas Earthquake Swarm. doi:10.7914/SN/X9_2008
- **XA_2008**: Anne Trehu, Mark Williams (2008): Monitgoring seismicity associated with a possible asperity on the Cascadia megathrust. doi:10.7914/SN/XA_2008
- **XB_2009**: Alan Levander, Gene Humphreys, Pat Ryan (2009): Program to Investigate Convective Alboran Sea System Overturn. doi:10.7914/SN/XB_2009
- **XB_2014**: Robert Woodward, Dan Hollis, Neil Spriggs (2014): Sweetwater Array. doi:10.7914/SN/XB_2014
- **XC_2006**: David James, Matthew Fouch (2006): Collaborative Research: Understanding the causes of continental intraplate tectonomagmatism: A case study in the Pacific Northwest. doi:10.7914/SN/XC_2006
- **XD_2011**: Simon Klemperer (2011): Passive seismic study of a magma-dominated rift: the Salton Trough. doi:10.7914/SN/XD_2011
- **XD_2014**: Ken Creager (2014): Collaborative Research: Illuminating the architecture of the greater Mount St. Helens magmatic systems from slab to surface. doi:10.7914/SN/XD_2014
- **XE_2005**: Tom Owens, Craig Jones (2005): Sierra Nevada EarthScope Project. doi:10.7914/SN/XE_2005
- **XE_2012**: Paul Bodin (2012): Seismic Activity of Low Angle Normal Faults in Death Valley, Caifornia. doi:10.7914/SN/XE_2012

- **XF_2006**: Steve Grand, Jim Ni (2006): Mapping the Rivera Subduction Zone. doi:10.7914/SN/XF_2006
- **XG_2009**: Ken Creager, John Vidale, Steve Malone (2009): Cascadia Array of Arrays. doi:10.7914/SN/XG_2009
- **XI_2011**: Suzan van der Lee, Douglas Wiens, Michael Wysession, Justin Revenaugh, Seth Stein, Andrew Frederiksen, Fiona Darbyshire, Donna Jurdy, Patrick Shore (2011): Superior Province Rifting Earthscope Experiment. doi:10.7914/SN/XI_2011
- **XI_2014**: Mitchell Barklage (2014): Long Beach Broadband or LA Syncline Seismic Interferometry Experiment. doi:10.7914/SN/XI_2014
- **XJ_2008**: Glenn Biasi (2008): Wells, Nevada Aftershock Recording. doi:10.7914/SN/XJ_2008
- **XN_2008**: Alan Levander (2008): Bolivar: Western Venezuela. doi:10.7914/SN/XN_2008
- **XN_2010**: Anne Trehu, Mark Williams (2010): Monitgoring seismicity associated with a possible asperity on the Cascadia megathrust. doi:10.7914/SN/XN_2010
- **XN_2011**: Kasper van Wijk (2011): joint summer field camp BSU/CSM/Imperial College of London. doi:10.7914/SN/XN_2011
- **XO_2011**: Gary Pavlis, Hersh Gilbert (2011): Ozark Illinois Indiana Kentucky (OIINK) Flexible Array Experiment. doi:10.7914/SN/XO_2011
- **XP_2008**: Ken Dueker, Rick Aster (2008): Colorado Rockies Experiment and Seismic Transect. doi:10.7914/SN/XP_2008
- **XQ_2006**: Yong-Gang Li, John Vidale (2006): Collaborative Research: Understanding Fault Zone Compliance by Seismic Probing of InSAR Anomalies. doi:10.7914/SN/XQ_2006
- **XQ_2007**: Alan Levander (2007): Seismic and Geodetic Investigations of Mendocino Triple Junction Dynamics. doi:10.7914/SN/XQ_2007
- **XQ_2012**: Lara Wagner (2012): Pre-Hydrofracking Regional Assessment of Central Carolina Seismicity. doi:10.7914/SN/XQ_2012
- **XR_2008**: Jay Pulliam, Steve Grand, Judy Sansom (2008): Seismic Investigation of Edge Driven Convection Associated with the Rio Grande Rift. doi:10.7914/SN/XR_2008
- **XT_2009**: Gregory P Waite (2009): Fuego Volcano 2009. doi:10.7914/SN/XT_2009
- **XT_2011**: Ray Russo (2011): Western Idaho Shear Zone - Passive. doi:10.7914/SN/XT_2011
- **XU_2006**: Steve Malone, Ken Creager, Stephane Rondenay, Tim Melbourne, Geoffrey Abers (2006): Collaborative Research: Earthscope integrated investigations of Cascadia subduction zone tremor, structure and process. doi:10.7914/SN/XU_2006
- **XU_2014**: Anne Sheehan (2014): Greeley Colorado RAMP Deployment 2014. doi:10.7914/SN/XU_2014
- **XU_2016**: Anne Sheehan (2016): USGS NEHRP Proposal 2016-0180 - Greeley. doi:10.7914/SN/XU_2016
- **XV_2009**: Anne Sheehan, Kate Miller, Megan Anderson, Christine Smith Siddoway, Eric Erslev (2009): Collaborative Research: Geometry and kinematics of basement-involved foreland arches:Insights into continental processes from Earthscope. doi:10.7914/SN/XV_2009
- **XY_2005**: Ken Dueker, George Zandt (2005): Magma Accretion and the Formation of Batholiths. doi:10.7914/SN/XY_2005
- **XY_2011**: Martin Chapman (2011): Experiment to Determine Hypocenters and Focal Mechanisms of Earthquakes Occurring in Association with Imaged Faults Near Summeville, South Carolina. doi:10.7914/SN/XY_2011
- **Y2_2013**: University of Bristol (2013): Balcombe. doi:10.7914/SN/Y2_2013
- **Y3_2008**: Glenn Biasi (2008): Wells, Nevada Aftershock Recording. doi:10.7914/SN/Y3_2008
- **Y7_2009**: Kasper van_Wijk, Mike Batzle, Lee Liberty, Andre Revil (2009): Geothermal Exploration Arkansas Valley. doi:10.7914/SN/Y7_2009
- **Y8_2009**: Charles Langston, Heather DeShon (2009): Detection and location of non-volcanic tremor in the New Madrid Seismic Zone. doi:10.7914/SN/Y8_2009
- **YB_2005**: Gregory P Waite (2005): Mount Saint Helens Dense Array. doi:10.7914/SN/YB_2005
- **YC_2011**: Anne Meltzer (2011): RAMP Virginia. doi:10.7914/SN/YC_2011
- **YD_2008**: John Louie (2008): Recording Mogul Events throughout the Reno Basin. doi:10.7914/SN/YD_2008
- **YE_2008**: Jamie Steidl (2008): Title: NEESR-SG: High Fidelity site characterization by experimentation, field observation, and inversion-based modeling. doi:10.7914/SN/YE_2008
- **YE_2011**: Bruce Beaudoin, Tim Parker, Eliana Arias-Dotson (2011): Testing TA & FA vaults and directly buried sensor (3T). doi:10.7914/SN/YE_2011
- **YG_2012**: Anne Trehu, Geoffrey Abers (2012): Collaborative Research: Imaging the Cascadia subduction zone - a ship-to-shore opportunity. doi:10.7914/SN/YG_2012
- **YH_2008**: Rebecca Saltzer, Gene Humphreys (2008): LaBarge Experiment. doi:10.7914/SN/YH_2008
- **YO_2014**: Gaherty, James B., Laura Wagner, Anne Becel, Margaret Benoit, Maureen Long, Donna Shillington, Harm Van Avendonck, Brandon Dugan (2014): Eastern North American Margin Community Seismic Experiment. doi:10.7914/SN/YO_2014
- **YQ_2009**: David Hawthorn (2009): Lincoln Noise Study. doi:10.7914/SN/YQ_2009
- **YW_2005**: Ken Creager (2005): Stalking Cascadia episodic tremor and slip with enhanced GPS and seismic arrays. doi:10.7914/SN/YW_2005
- **YW_2016**: Kent Anderson, Justin Sweet, Bob Woodward (2016): IRIS Community Wavefield Experiment in Oklahoma. Incorporated Research Institutions for Seismology. doi:10.7914/SN/YW_2016
- **YX_2008**: James Cochran, Yong-Gang Li, Jamie Steidl (2008): Seismology Rapid Response Test During the SoSAF Shakeout. doi:10.7914/SN/YX_2008
- **YX_2010**: Simon Klemperer, Kate Miller (2010): Collaborative Research: 4D multi-disciplinary investigation of highly variable crustal response to continental extension in the north-central Basin and Range. doi:10.7914/SN/YX_2010
- **YY_2012**: Jake Walter (2012): Mendenhall Glacier Outburst Flood Seismicity and Next-Generation Instrument Testing. doi:10.7914/SN/YY_2012
- **YZ_2009**: Susan Y. Schwartz, Andrew Newman, Marino Protti, Victor Gonzalez (2009): Nicoya Seismogenic Zone. doi:10.7914/SN/YZ_2009
- **Z3_2009**: Silver (2009): Detecting Structural changes During an ETS Event: Proof of Concept. doi:10.7914/SN/Z3_2009
- **Z3_2010**: Shari Kelley, Greg Kaufman, Michael Albrecht (2010): Jemez Pueblo geothermal project. doi:10.7914/SN/Z3_2010
- **Z5_2013**: John Nabelek, Jochen Braunmiller (2013): Seismicity, Structure and Dynamics of the Gorda Deformation Zone. doi:10.7914/SN/Z5_2013
- **Z9_2010**: Karen M. Fischer, Robert B. Hawman, Lara S. Wagner (2010): Southeastern Suture of the Appalachian Margin Experiment. doi:10.7914/SN/Z9_2010
- **ZA_2006**: Michael West (2006): The Colima Deep Seismic Experiment: Imaging the Magmatic Root of Colima Volcano. doi:10.7914/SN/ZA_2006

- **ZH_2010**: Anne Sheehan, Kate Miller, Megan Anderson, Christine Smith Siddoway, Eric Erslev (2010): Collaborative Research: Geometry and kinematics of basement-involved foreland arches:Insights into continental processes from Earthscope. doi:10.7914/SN/ZH_2010
- **ZH_2011**: Anne Trehu, Mark Williams (2011): Monitoring seismicity associated with a possible asperity on the Cascadia mega-thrust. doi:10.7914/SN/ZH_2011
- **ZI_2010**: Kate Miller, Anne Sheehan, Megan Anderson, Christine Smith Siddoway, Eric Erslev, Steven Harder (2010): Collaborative Research: Geometry and kinematics of basement-involved foreland arches: Insights into continental processes from Earthscope. doi:10.7914/SN/ZI_2010
- **ZO_2010**: Diana Roman, Peter La_Femina (2010): Volcanic stress field analysis using non-local seismic sources at Hekla Volcano, Iceland. doi:10.7914/SN/ZO_2010
- **ZQ_2013**: Harold Gurrola (2013): Imaging the Matador arch using receiver functions from Texan dataloggers and short period geophones. doi:10.7914/SN/ZQ_2013
- **ZS_2009**: Sridhar Anandakrishnan (2009): CReSIS. doi:10.7914/SN/ZS_2009
- **ZW_2013**: Heather DeShon, Chris Hayward, Brian Stump, M. Beatrice Magnani, Matthew Hornbach (2013): North Texas Earthquake Study: Azle and Irving/Dallas. doi:10.7914/SN/ZW_2013
- **ZZ_2012**: Anne Trehu, Geoffrey Abers (2012): Collaborative Research: Imaging the Cascadia subduction zone - a ship-to-shore opportunity. doi:10.7914/SN/ZZ_2012

Temporary deployments without an assigned DOI at the time of writing that were utilized for this chapter: 2D_2010, 4E_2010, 4F_2007, 6A_2008, 7A_2010, 7F_2010, 8A_2010, X4_2010, X5_2007, X9_2012, XG_2014, XI_2005, XT_2006, XW_2013, XY_2013, Y9_2009, YB_2010, YC_2013, YF_2006, YF_2010, YY_2005, Z2_2006, Z2_2009, Z3_2008, Z4_2009, ZH_2016, ZN_2006, and ZX_2005.

6
Conclusion & Outlook

In this work we motivated, developed, and described technical advances to enable the reliable usage of current data volumes in seismology to incorporate them into full seismic waveform inversions using the adjoint-state method before applying these techniques to invert for a new seismic velocity model of the subsurface beneath North America and the Northern Atlantic.

The ObsPy toolkit is described in chapter 2. It is a Python toolkit for seismology capable of reading and writing essentially every file format currently in use in seismology. Additionally, it can acquire data from data centers and network operators around the globe and it offers flexible signal processing functionality in a language understandable by seismologists. The key feature of ObsPy for many of the developments that follow is that it is written in Python, a general purpose programming language that just happens to be very useful for scientific work. This sparked a very large and versatile ecosystem with high-quality packages from all branches of science that we can directly make use of in many cases. Additionally, and in contrast to scientific packages like Matlab, it also offers lots of IT tools like proper database adapters, web capabilities, and development environments. In the end, this means that it is possible to create workflows and utilities and tie together smaller components to more complex ones tackling larger problems. This would have been much harder or impossible without the development and availability of ObsPy.

Chapter 3 shows this on the concrete example of LASIF, the Large-scale Seismic Inversion Framework. Full seismic waveform inversions are an inherently iterative process slowly changing and improving an initial Earth model towards a model that better explains the observed waveforms. This for one requires access to waveform data possibly scattered at data centers around the world. Furthermore, it necessitates information about the used seismic receivers and the actual earthquakes. This is true for most branches of seismology and ObsPy handles that to some extent. Waveform inversions moreover call for choosing windows in which observed and synthetic data is similar enough, for misfits and adjoint sources to be calculated, for keeping track of which earthquake has been recorded at which receiver and if that particular waveform has a satisfactory signal to noise ratio, for knowing if recordings were used in a particular iteration, and a slew of other things. All of these pieces of information are in some kind of relation to each other and might depend on and influence other pieces of data. Manually handling this when hundreds of thousands of waveforms are used over tens of iterations becomes impossible and especially error prone in subtle ways that might not be discovered but have an influence on the final model. The LASIF framework has been designed to tackle this and we showed some of the chosen approaches and also applied it to a full seismic waveform inversion for Japan.

The experience gathered in developing and using ObsPy as well as LASIF led to the conclusion that none of the seismological data formats currently in existence are satisfactory for our purposes. They have not been designed with modern architectures and workflows in mind

and have limitations regarding raw efficiency due to no possibility of using parallel I/O and the imposed necessary high file count. Furthermore, they cannot make up complete data sets containing many pieces requiring complex and custom file system layouts that change from project to project. This inhibits data collaboration and exchange and ultimately the reproducibility of scientific results. The proposed ASDF data format in chapter 4 offers a solution to these problem while at the same time being practically useful and fairly simple in itself. It manages to do so by building on existing solutions wherever possible without reinventing the wheel. Data is stored with a fixed layout inside an HDF5 file. Earthquakes are stored in the existing QuakeML format whereas station information is stored with the familiar FDSN StationXML format, both directly in the HDF5 file. Additionally, ASDF can take care of non-waveform data and keep a provenance record for each piece of information, if so desired. To foster acceptance and adoption in the community we created extensive specification and documentation as well as packages for Python and C to make it directly usable.

In order to assure that these developments actually work as intended and to prevent their faith from being pure programming exercises we performed a large-scale full seismic waveform inversion for subsurface structure beneath North America and the Northern Atlantic in chapter 5. The USArray project guaranteed a lot of available high-quality data recordings in the chosen inversion domain. A spectral element waveform propagation code was used for the forward as well as the adjoint waveform simulations. All in all, we performed 20 L-BFGS iteration steps, yielding a radially anisotropic elastic model that explains seismic waveforms with periods from 30 to about 120 seconds. Later iterations used 72 events with about 150‘000 unique and usually three components records resulting in on the order of one million time-frequency phase misfit measurements per iteration. These amounts of data required some additional workflow developments. To establish trust in the final model we used ten validation events that were not inverted for. With an independent and to some degree worst case full seismogram L^2-distance misfit measurement seismograms improved in 75% of all cases when simulating with the final model compared to the initial model. Additionally, we carried out some quantitative resolution checks to gauge the spatially varying resolution of the model. The final model generates seismograms that in many cases can reproduce full observed seismograms including body and complex surface waves. An added complexity of the procedure was the inclusion of a high-resolution subregion originating from an existing full waveform inversion in the starting model. The inversion expanded features already visible in that region and continued them into the low-resolution region of the initial model.

Future works will need to attack on multiple angles. The model resolution could likely be pushed higher by a combination of more accurate simulations of the physics including so far ignored effects, using even more data, and the adaption of more sophisticated weighting schemes and misfit definitions as debated in chapter 5.6. However, it will also require the continued development of and investment in the underlying codes and tools to even enable that in the first place. We will soon live in an exa-scale computing world with likely very heterogeneous computer architectures. These all will require specialized knowledge to make the most of. In turn this will likely lead to a further need for people working at the interface of seismology, computer science, and engineering and who are capable and willing to develop tools and applications to empower the rest of the community.

Bibliography

Afanasiev, M., Peter, D., Sager, K., Simute, S., Ermert, L., Krischer, L., and Fichtner, A. (2016). Foundations for a multiscale collaborative Earth model. *Geophysical Journal International*, **204**(1):39–58. doi:10.1093/gji/ggv439.
Referenced in: 5, 5.2.1

Afanasiev, M., Pratt, R. G., Kamei, R., and McDowell, G. (2014). Waveform-based simulated annealing of crosshole transmission data: A semi-global method for estimating seismic anisotropy. *Geophys. J. Int.*, **199**:1586–1607.
Referenced in: 3.1, 3.7

Aki, K., Christoffersson, A., and Husebye, E. S. (1977). Determination of the three-dimensional seismic structure of the lithosphere. *J. Geophys. Res.*, **82**:227–296.
Referenced in: 1.1, 3.1, 5.1

Aki, K. and Lee, W. H. K. (1976). Determination of three-dimensional velocity anomalies under a seismic array using first P arrival times from local earthquakes - 1. A homogeneous initial model. *J. Geophys. Res.*, **81**:4381–4399.
Referenced in: 3.1

Al-Attar, D. and Woodhouse, J. H. (2008). Calculation of seismic displacement fields in self-gravitating earth models - applications of minors vectors and symplectic structure. *Geophys. J. Int.*, **175**(3):1176–1208. doi:10.1111/j.1365-246X.2008.03961.x.
Referenced in: A.1, A.4.1, A.5.3

Apache Software Foundation (2016a). Hadoop. `https://hadoop.apache.org`.
Referenced in: 4.5.2

Apache Software Foundation (2016b). Spark. `https://spark.apache.org`.
Referenced in: 4.5.2

Astiz, L., Earle, P., and Shearer, P. (1996). Global Stacking of Broadband Seismograms. *Seismol. Res. Lett.*, **67**(4):8–18. doi:10.1785/gssrl.67.4.8.
Referenced in: A.1

Atkinson, M., Baxter, R., Brezany, P., Corcho, O., Galea, M., Parsons, M., Snelling, D., and van Hemert, J. (2013). *The Data Bonanza: Improving Knowledge Discovery in Science, Engineering, and Business*. John Wiley & Sons, Hoboken, New Jersey.
Referenced in: 2.5

Auer, L., Boschi, L., Becker, T. W., Nissen-Meyer, T., and Giardini, D. (2014). Savani : A variable resolution whole-mantle model of anisotropic shear velocity variations based on multiple data sets. *J. Geophys. Res. Solid Earth*, **119**(4):3006–3034. doi:10.1002/2013JB010773.
Referenced in: A.5.3

Backus, G. E. (1962). Long-Wave Elastic Anisotropy Produced by Horizontal Layering. *Journal of Geophysical Research*, **67**(11):4427–4440. doi:10.1029/JZ067i011p04427.
Referenced in: 5.5.3

Banerdt, W. (2013). InSight: A Discovery Mission to Explore the Interior of Mars. *44th Lunar Planet. Sci. Conf.*
Referenced in: A.5.6

Basini, P., Nissen-Meyer, T., Boschi, L., Casarotti, E., Verbeke, J., Schenk, O., and Giardini, D. (2013). The influence of nonuniform ambient noise on crustal tomography in Europe. *Geochem. Geophys. Geosys.*, **14**(5):1471–1492. doi:10.1002/ggge.20081.
Referenced in: A.5.7, B.3.1

Bensen, G. D., Ritzwoller, M. H., Barmin, M. P., Levshin, A. L., Lin, F., Moschetti, M. P., Shapiro, N. M., and Yang, Y. (2007). Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements. *Geophys. J. Int.*, **169**:1239–1260.
Referenced in: 3.11

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., and Wassermann, J. (2010). ObsPy: A Python Toolbox for Seismology. *Seismological Research Letters*, **81**(3):530.
Referenced in: 2, 2.1, 3.1, 3.3, 4.1.1, 4.5.5, 5.1, A.3, A.6, B.3.3

Bird, I., Buncic, P., Carminati, F., Cattaneo, M., Clarke, P., Fisk, I., Girone, M., Harvey, J., Kersevan, B., Mato, P., Mount, R., and Panzer-Steindel, B. (2014). Update of the Computing Models of the WLCG and the LHC Experiments. *Technical Report CERN-LHCC-2014-014 / LCG-TDR-002*, CERN, Geneva.
Referenced in: 4.2.1

Bird, P. (2003). An updated digital model of plate boundaries. *Geochemistry, Geophysics, Geosystems*, **4**(3). doi:10.1029/2001GC000252.
Referenced in: 3.10, 5.1

Blom, N., Boehm, C., and Fichtner, A. (2017). Synthetic inversions for density using seismic and gravity data. *Geophysical Journal International*. Accepted.
Referenced in: 5.2.2

Boehm, C., Hanzich, M., de la Puente, J., and Fichtner, A. (2016). Wavefield compression for adjoint methods in full-waveform inversion. *Geophysics*, **81**(6). doi:10.1190/geo2015-0653.1.
Referenced in: 5.3

Bormann, P. (2012). *New Manual of Seismological Observatory Practice (NMSOP-2)*. IASPEI, GFZ German Research Centre for Geosciences, Potsdam. doi:10.2312/GFZ.NMSOP-2. `http://nmsop.gfz-potsdam.de/`.
Referenced in: 4.3

Bozdağ, E., Peter, D., Lefebvre, M., Komatitsch, D., Tromp, J., Hill, J., Podhorszki, N., and Pugmire, D. (2016). Global Adjoint Tomography: First-Generation Model. *Geophysical Journal International*. doi:10.1093/gji/ggw356.
Referenced in: 1.2, 5.1, 5.3, 5.5.3

Bozdağ, E., Trampert, J., and Tromp, J. (2011). Misfit functions for full waveform inversion based on instantaneous phase and envelope measurements. *Geophysical Journal International*, **185**(2):845–870. doi:10.1111/j.1365-246X.2011.04970.x.
Referenced in: 5.2.3

Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and Yergeau, F. (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition). Accessed 7-October-2015, `https://www.w3.org/TR/2008/REC-xml-20081126/`.
Referenced in: 4.2.2

Brown, J., Knepley, M., and Smith, B. (2015). Run-Time Extensibility and Librarization of Simulation Software. *Computing in Science & Engineering*, **17**(1):38–45. doi:10.1109/MCSE.2014.95.
Referenced in: B.5.1

Buland, R. and Chapman, C. (1983). The computation of seismic travel times. *Bulletin of the Seismological Society of America*, **73**(5):1271–1302.
Referenced in: 2.3.1, B.3.3

Burger, W. and Burge, M. (2009). *Principles of Digital Image Processing: Core Algorithms*. Springer.
Referenced in: A.2.4, A.2.4

Capdeville, Y. (2010). 1-D non-periodic homogenization for the seismic wave equation. *Geophysical Journal …*, pages 897–910. doi:10.1111/j.1365-246X.2010.04529.x.
Referenced in: 5.5.3

Ceylan, S., van Driel, M., Euchner, F., Khan, A., Clinton, J., Krischer, L., Böse, M., and Giardini, D. (in review). From initial models of seismicity, structure and noise to synthetic seismograms for Mars. *Space Sci. Rev.*
Referenced in: B.6

Chapman, C. H. (1978). A new method for computing synthetic seismograms. *Geophys. J. Int.*, **54**(3):481–518. doi:10.1111/j.1365-246X.1978.tb05491.x.
Referenced in: A.1, A.5.3

Chen, P., Zhao, L., and Jordan, T. H. (2007). Full 3D tomography for the crustal structure of the Los Angeles region. *Bulletin of the Seismological Society of America*, **97**:1094–1120.
Referenced in: 3.1

Colli, L., Fichtner, A., and Bunge, H.-P. (2013). Full waveform tomography of the upper mantle in the South Atlantic region: Imaging a westward fluxing shallow asthenosphere? *Tectonophysics*, **604**:26–40. doi:10.1016/j.tecto.2013.06.015.
Referenced in: 4.5.5, 5, 5.2.1, 5.2, 5.3, 5.5.4

Colombi, A., Nissen-Meyer, T., Boschi, L., and Giardini, D. (2014). Seismic waveform inversion for core-mantle boundary topography. *Geophys. J. Int.*, **198**(1):55–71. doi:10.1093/gji/ggu112.
Referenced in: A.5.3

Crotwell, H. P., Owens, T. J., and Ritsema, J. (1999). The TauP Toolkit: Flexible seismic travel-time and ray-path utilities. *Seismological Research Letters*, **70**(2):154–160.
Referenced in: 2.3.1, 2.2, A.1, B.3.3

Dahlen, F., Hung, S.-H., and Nolet, G. (2000). Fréchet kernels for finite-frequency traveltimes – I. Theory. *Geophysical Journal International*, **141**:157–174.
Referenced in: 1.1, 3.1, 5.1

De la Cruz-Reyna, S. and Reyes-Dávila, G. (2001). A model to describe precursory material-failure phenomena: applications to short-term forecasting at colima volcano, Mexico. *Bulletin of Volcanology*, **63**(5):297–308.
Referenced in: 2.4.3

De la Cruz-Reyna, S., Tárraga, M., Ortiz, R., and Martínez-Bringas, A. (2010). Tectonic earthquakes triggering volcanic seismicity and eruptions. Case studies at Tungurahua and Popocatépetl volcanoes. *Journal of Volcanology and Geothermal Research*, **193**(1):37–48.
Referenced in: 2.4.3

Deichmann, N. and Garcia-Fernandez, M. (1992). Rupture geometry from high-precision relative hypocentre locations of microearthquake clusters. *Geophysical Journal International*, **110**(3):501–517.
Referenced in: B.5.2

Di Grazia, G., Falsaperla, S., and Langer, H. (2006). Volcanic tremor location during the 2004 Mount Etna lava effusion. *Geophysical Research Letters*, **33**(4).
Referenced in: 2.4.3

Díaz, J., Villaseñor, A., Gallart, J., Morales, J., Pazos, A., Córdoba, D., Pulgar, J., García-Lobón, J. L., Harnafi, M., and TopoIberia Seismic Working Group (2009). The IBER-ARRAY broadband seismic network: A new tool to investigate the deep structure beneath Iberia. *ORFEUS Newsletter*, **8**:1–6.
Referenced in: 3.1

Diaz-Steptoe, H. (2013). *Full seismic waveform tomography of the Japan region using adjoint methods*. Phd thesis, Utrecht University.
Referenced in: 3.10, 3.8

Dziewonski, A. and Romanowicz, B. (2007). Seismology and the Structure of the Earth. *Treatise on Geophysics 1*.
Referenced in: 1, 1.1

Dziewonski, A. M. and Anderson, D. L. (1981). Preliminary reference Earth model. *Physics of the Earth and Planetary Interiors*, **25**(4):297–356. doi:10.1016/0031-9201(81)90046-7.
Referenced in: 1.1, 5.2.1, 5.2.2, 5.13, 5.5.4, B.4

Dziewoński, A. M., Hager, B. H., and O'Connell, R. J. (1977). Large-scale heterogeneities in the lower mantle. *Journal of Geophysical Research*, **82**:239–255.
Referenced in: 3.1, 5.1

Ekström, G., Nettles, M., and a.M. Dziewoński (2012). The global CMT project 2004–2010: Centroid-moment tensors for 13,017 earthquakes. *Physics of the Earth and Planetary Interiors*, **200-201**:1–9.
Referenced in: 3.3, 5.4, 5.5.3, 5.6, B.3.1, B.3.4

Endo, E. and Murray, T. (1991). Real-time Seismic Amplitude Measurement (RSAM): a volcano monitoring and prediction tool. *Bulletin of Volcanology*, **53**(7):533–545.
Referenced in: 2.4.3

Evans, P. L., Strollo, A., Clark, A., Ahern, T., Newman, R., Clinton, J. F., Pedersen, H., and Pequegnat, C. (2015). Why seismic networks need digital object identifiers. *Eos*, **96**. doi: 10.1029/2015EO036971.
Referenced in: 5.6

Faccioli, E., Maggio, F., Paolucci, R., and Quarteroni, a. (1997). 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method. *Journal of Seismology*, **1**(3):237–251. doi:10.1023/a:1009758820546.
Referenced in: 5.2.1

Fichtner, A. (2011). *Full Seismic Waveform Modelling and Inversion*. Springer Berlin Heidelberg.
Referenced in: 5.2, 5.2.3, 5.2.4

Fichtner, A., Bunge, H.-P., and Igel, H. (2006). The adjoint method in seismology I. Theory. *Physics of the Earth and Planetary Interiors*, **157**(1-2):86–104. doi:10.1016/j.pepi.2006.03.016.
Referenced in: 1.2, 3.1, 3.7, 4.5.5, 5.1, 5.2.2, 5.3

Fichtner, A. and Igel, H. (2008). Efficient numerical surface wave propagation through the optimization of discrete crustal models - a technique based on non-linear dispersion curve matching (DCM). *Geophysical Journal International*, **173**:519–533.
Referenced in: 3.5, 3.10, 5.5.3

Fichtner, A., Kennett, B. L. N., Igel, H., and Bunge, H.-P. (2008). Theoretical background for continental- and global-scale full-waveform inversion in the time-frequency domain. *Geophysical Journal International*, **175**:665–685.
Referenced in: 3.7, 3.10

Fichtner, A., Kennett, B. L. N., Igel, H., and Bunge, H.-P. (2009). Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods. *Geophysical Journal International*, **179**:1703–1725.
Referenced in: 1.2, 3.1, 3.5, 3.10, 5.1, 5.2, 5.2.3, 5.3

Fichtner, A. and Leeuwen, T. V. (2015). Resolution analysis by random probing. *Journal of Geophysical Research: Solid Earth*, **120**(8):5549–5573. doi:10.1002/2015JB012106.
Referenced in: 5.5.2, 5.10

Fichtner, A., Trampert, J., Cupillard, P., Saygin, E., Taymaz, T., Capdeville, Y., and Villasenor, A. (2013). Multiscale full waveform inversion. *Geophysical Journal International*,

pages 534–556.
Referenced in: 3.1, 3.6, 3.7, 5, 5.2.1, 5.2.2, 5.2, 5.5.2, 5.5.4

Fichtner, A. and van Driel, M. (2014). Models and Frechet kernels for frequency-(in)dependent Q. *Geophysical Journal International*, **198**(3):1878–1889.
Referenced in: 2.5

Friederich, W. (2003). The S-velocity structure of the East Asian mantle from inversion of shear and surface waveforms. *Geophys. J. Int.*, **153**:88–102.
Referenced in: 1.1, 3.1, 5.1

Friederich, W. and Dalkolmo, J. (1995). Complete synthetic seismograms for a spherically symmetric earth by a numerical computation of the Green's function in the frequency domain. *Geophys. J. Int.*, **122**(2):537–550. doi:10.1111/j.1365-246X.1995.tb07012.x.
Referenced in: A.1

Froment, B., Campillo, M., Roux, P., Gouédard, P., Verdel, A., and Weaver, R. L. (2010). Estimation of the effect of nonisotropically distributed energy on the apparent arrival time in correlations. *Geophysics*, **75**(5):SA85.
Referenced in: A.5.7

Fuchs, F., Kolínský, P., Gröschl, G., Bokelmann, G., and the AlpArray Working Group (2016). AlpArray in Austria and Slovakia: technical realization, site description and noise characterization. *Advances in Geosciences*, **43**:1–13. doi:10.5194/adgeo-43-1-2016.
Referenced in: 1.1

Fuchs, K. and Müller, G. (1971). Computation of synthetic seismograms with the reflectivity method and comparison with observations. *Geophys. J. R. Astron. Soc.*, **23**:417–433.
Referenced in: A.1, A.5.3

Gee, L. S. and Jordan, T. H. (1992). Generalized seismological data functionals. *Geophys. J. Int.*, **111**:363–390.
Referenced in: 3.7

Gee, L. S. and Leith, W. S. (2011). The Global Seismographic Network. *U.S. Geological Fact Sheet*, **2011-3021**.
Referenced in: 3.1

Geller, R. J. and Ohminato, T. (1994). Computation of synthetic seismograms and their partial derivatives for heterogeneous media with arbitrary natural boundary conditions using the Direct Solution Method. *Geophys. J. Int.*, pages 421–446. doi:10.1111/j.1365-246X.1994.tb01807.x.
Referenced in: A.1

Gokhberg, A. and Fichtner, A. (2016). Full-waveform inversion on heterogeneous HPC systems. *Computers & Geosciences*, pages 1–9. doi:10.1016/j.cageo.2015.12.013.
Referenced in: 5.2.1

Gouédard, P., Stehly, L., Brenguier, F., Campillo, M., Colin de Verdière, Y., Larose, E., Margerin, L., Roux, P., Sánchez-Sesma, F. J., Shapiro, N. M., and Weaver, R. L. (2008).

Cross-correlation of random fields: mathematical approach and applications. *Geophys. Prospect.*, **56**(3):375–393. doi:10.1111/j.1365-2478.2007.00684.x.
Referenced in: A.5.7

Govoni, A., Bonatto, L., Capello, M., Cavaliere, A., Chiarabba, C., D'Alema, E., Danesi, S., Lovati, S., Margheriti, L., Massa, M., Mazza, S., Mazzarini, F., Monna, S., Moretti, M., Nardi, A., Piccinini, D., Piromallo, C., Pondrelli, S., Salimbeni, S., Serpelloni, E., Solarino, S., Vallocchia, M., Santulin, M., and the AlpArray Working Group (2017). AlpArray-Italy: Site description and noise characterization. *Advances in Geosciences*, **43**:39–52. doi:10.5194/adgeo-43-39-2017.
Referenced in: 1.1

Grand, S., VanDerHilst, R., and Widiyantoro, S. (1997). Global seismic tomography: A snapshot of convection in the Earth. *Geol. Soc. Am. Today*, **7, No.4**:1–7.
Referenced in: 3.1, 5.1

Gualtieri, L., Stutzmann, E., Capdeville, Y., Ardhuin, F., Schimmel, M., Mangeney, a., and Morelli, a. (2013). Modelling secondary microseismic noise by normal mode summation. *Geophys. J. Int.*, **193**(3):1732–1745. doi:10.1093/gji/ggt090.
Referenced in: A.5.7

Gutenberg, B. (1913). Über die Konstitution des Erdinneren, erschlossen aus Erdbebenbeobachtungen. *Physikalische Zeitschrift*, **14**:1217–1218.
Referenced in: 1.1

Gutenberg, B. (1926). Untersuchungen zur Frage, bis zu welcher Tiefe die Erde kristallin ist. *Z. Geophys.*, **2**:24–29.
Referenced in: 1.1

Gutenberg, Beno (1959). *Physics of the Earth's Interior*. Academic Press Inc.
Referenced in: 1

Hadziioannou, C., Gaebler, P., Schreiber, U., Wassermann, J., and Igel, H. (2012). Examining ambient noise using colocated measurements of rotational and translational motion. *Journal of Seismology*, **16**(4):787–796.
Referenced in: 2.5

Halley, E. (1686). An Account of the Cause of the Change of the Variation of the Magnetical Needle; With an Hypothesis of the Structure of the Internal Parts of the Earth: As It Was Proposed to the Royal Society in One of Their Late Meetings. By Edm. Halley. *Proceedings of the Royal Society of London*, **16**(179-191):563–578. doi:10.1098/rstl.1686.0107.
Referenced in: 1.1

Havskov, J. (2010). *Routine data processing in earthquake seismology with sample data, exercises and software*. Springer, Dordrecht New York. ISBN: 978-90-481-8696-9.
Referenced in: 4.3

Helffrich, G., Wookey, J., and Bastow, I. (2013). *The Seismic Analysis Code: A Primer and User's Guide*. Cambridge University Press, 1 edition.
Referenced in: 4.3.2, 4.5.5, B.3.1

Hjörleifsdóttir, V. and Ekström, G. (2010). Effects of three-dimensional Earth structure on CMT earthquake parameters. *Physics of the Earth and Planetary Interiors*, **179**:178–190. doi: 10.1016/j.pepi.2009.11.003.
Referenced in: 5.5.3

Holtzman, B., Candler, J., Turk, M., and Peter, D. (2013). Seismic Sound Lab: Sights, Sounds and Perception of the Earth as an Acoustic Space. M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. l. Ystad, editors, *Sound, Music. Motion*, pages 161–174. Springer, Marseille.
Referenced in: A.2.2

Hosseini, K. and Sigloch, K. (2015). Multi-frequency measurements of core-diffracted P-waves (Pdiff) for global waveform tomography. *Geophys. J. Int.* Submitted to Geophy. J. Int.
Referenced in: A.5.3

Hua, C. (1990). An inverse transformation for quadrilateral isoparametric elements: analysis and application. *Finite Elem. Anal. Des.*, **7**:159–166.
Referenced in: A.2.3

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, **9**(3):90–95. doi:10.1109/MCSE.2007.55.
Referenced in: 2.1

Igel, H., Djikpesse, H., and Tarantola, A. (1996). Waveform inversion of marine reflection seismograms for P impedance and Poisson's ratio. *Geophysical Journal International*, **124**:363–371.
Referenced in: 3.7

Igel, H., Takeuchi, N., Geller, R. J., Megnin, C., Bunge, H.-P., Clévédé, E., Dalkolmo, J., and Romanowicz, B. (2000). The COSY Project: verification of global seismic modeling algorithms. *Phys. Earth Planet. Inter.*, **119**:3–23. doi:10.1016/S0031-9201(99)00150-8.
Referenced in: A.1

Incorporated Research Institutions for Seismology (IRIS) (2012). *SEED Reference Manual - Standard for the Exchange of Earthquake Data*. `https://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf`.
Referenced in: 2.2.1, 2.2.2, 2.3.2, 4.2.3, B.3.1

Instrumental Software Technologies, Inc. (2014). JEvalResp. `http://www.isti.com/JEvalResp/`.
Referenced in: 2.3

IRIS (2014). IRIS Mustang Beta. *Online*. Accessed: 2014-07-03.
Referenced in: 3.6.2

IRIS DMC (2014a). Software Downloads: evalresp. `http://www.iris.edu/dms/nodes/dmc/software/downloads/evapresp/`.
Referenced in: 2.3.2

IRIS DMC (2014b). Software Downloads: rdseed. `http://www.iris.edu/dms/nodes/dmc/software/downloads/rdseed/`.
Referenced in: 2.3.2

IRIS/PASSCAL Data Group (2012). *Introduction to Active Source Data Archiving Utilizing PH5 as the Archive Format*. IRIS/PASSCAL Instrument Center. Version: 2012336.
Referenced in: 4.3.3

Ishii, M., Shearer, P. M., Houston, H., and Vidale, J. E. (2005). Extent, duration and speed of the 2004 Sumatra-Andaman earthquake imaged by the Hi-Net array. *Nature*, **435**(7044):933–6. doi:10.1038/nature03675.
Referenced in: B.5.5

Jeffreys, H. (1926). The Rigidity of the Earth's Central Core. *Monthly Notices of the Royal Astronomical Society, Geophysical Supplement*, pages 371–383.
Referenced in: 1.1

Jeffreys, H. and Bullen, K. E. (1940). Seismological Tables. *British Association for the Advancement of Science*.
Referenced in: 1.1

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed 2014-10-30], URL `http://www.scipy.org/`.
Referenced in: 2.1, 3.1, A.2.3, A.3

Kawai, K., Takeuchi, N., and Geller, R. J. (2006). Complete synthetic seismograms up to 2 Hz for transversely isotropic spherically symmetric media. *Geophys. J. Int.*, **164**(2):411–424. doi:10.1111/j.1365-246X.2005.02829.x.
Referenced in: A.1

Kennett, B. L. N. and Engdahl, E. R. (1991). Traveltimes for global earthquake location and phase identification. *Geophysical Journal International*, **105**(2):429–465.
Referenced in: 1.1, 2.3.1, B.4

Kennett, B. L. N., Engdahl, E. R., and Buland, R. (1995). Constraints on seismic velocities in the Earth from traveltimes. *Geophysical Journal International*, **122**:108–124.
Referenced in: 1.1, 3.6.1, 3.4, B.1, B.4

Khan, A. and Connolly, J. A. D. (2008). Constraining the composition and thermal state of Mars from inversion of geophysical data. *J. Geophys. Res.*, **113**(E7):E07003. doi:10.1029/2007JE002996.
Referenced in: A.5.6

Kikuchi, M. and Kanamori, H. (1982). Inversion of complex body waves. *Bull. Seismol. Soc. Am.*, **72**(2):491–506.
Referenced in: A.1, A.5.5, A.20

Kissling, E. (1988). Geotomography with local earthquake data. *Rev. Geophys.*, **26**:659–698.
Referenced in: 1.1, 3.1, 5.1

Knapmeyer, M. (2005). Numerical Accuracy of Travel-time Software in Comparison with Analytic Results. *Seismological Research Letters*, **76**(1):74–81.
Referenced in: 2.2

Komatitsch, D. (1997). *Méthodes spectrales et éléments spectraux pour l'équation de l'élastodynamique 2D et 3D en milieu hétérogène*. Ph.D. thesis, Institut de Physique du Globe de Paris.
Referenced in: 5.2.1

Komatitsch, D. and Tromp, J. (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, pages 806–822.
Referenced in: B.2.1

Komatitsch, D. and Tromp, J. (2002a). Spectral-element simulations of global seismic wave propagation – I. Validation. *Geophysical Journal International*, **149**:390–412.
Referenced in: 3.5, 4.1.1, 4.4.1, 4.5.5, A.2.3

Komatitsch, D. and Tromp, J. (2002b). Spectral-element simulations of global seismic wave propagation-II. Three-dimensional models, oceans, rotation and self-gravitation. *Geophysical Journal International*, **150**:308–318.
Referenced in: 3.5, 4.1.1, 4.4.1, 4.5.5, 5.5.3, A.1, B.2.1

Krischer, L. (2014). StationXML Test Case git repository. `https://github.com/obspy/sandbox/tree/master/stationxml_test`.
Referenced in: 2.3.2

Krischer, L., Fichtner, A., Zukauskaite, S., and Igel, H. (2015a). Large-Scale Seismic Inversion Framework. *Seismological Research Letters*, **86**(4):1198–1207. doi:10.1785/0220140248.
Referenced in: 5.1, 5.3

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., and Wassermann, J. (2015b). ObsPy: a bridge for seismology into the scientific Python ecosystem. *Computational Science & Discovery*, **8**(1):14003–14020. doi:10.1088/1749-4699/8/1/014003.
Referenced in: 3.1, 4.4.2, 5.1, B.2.2

Krischer, L., Smith, J., Lei, W., Lefebvre, M., Ruan, Y., Andrade, E. S. D., Podhorszki, N., Bozdağ, E., and Tromp, J. (2016). An Adaptable Seismic Data Format. *Geophysical Journal International*, **207**(2):1003–1011. doi:10.1093/gji/ggw319.
Referenced in: 5.1

Kristekova, M., Kristek, J., and Moczo, P. (2009). Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals. *Geophys. J. Int.*, **178**(2):813–825. doi:10.1111/j.1365-246X.2009.04177.x.
Referenced in: A.2.3, A.6, A.4.1, A.12

Laske, G. and Masters, G. (1996). Constraints on global phase velocity maps from long-period polarization data. *J. Geophys. Res.*, **101**:16059–16075.
Referenced in: 3.7

Lecocq, T., Caudron, C., and Brenguier, F. (2014). MSNoise, a Python Package for Monitoring Seismic Velocity Changes Using Ambient Seismic Noise. *Seismological Research Letters*, **85**(3):715–726. doi:10.1785/0220130073.
Referenced in: 2.4.3

Lehmann, I. (1936). P'. *Publications du Bureau Central Séismologique International.*, **A14**(3):87–115.
Referenced in: 1.1

Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, **45**(1-3):503–528. doi:10.1007/BF01589116.
Referenced in: 5.2.2

Liu, Q. and Gu, Y. (2012). Seismic imaging: From classical to adjoint tomography. *Tectonophysics*, **566-567**:31–66. doi:10.1016/j.tecto.2012.07.006.
Referenced in: 1.2, 4.5.5, 5.1

Liu, Q., Logan, J., Tian, Y., Abbasi, H., Podhorszki, N., Choi, J. Y., Klasky, S., Tchoua, R., Lofstead, J., Oldfield, R., Parashar, M., Samatova, N., Schwan, K., Shoshani, A., Wolf, M., Wu, K., and Yu, W. (2014). Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks. *Concurrency and Computation: Practice and Experience*, **26**(7):1453–1473. doi:10.1002/cpe.3125.
Referenced in: 4.2.1

Luo, Y. and Schuster, G. T. (1991). Wave-equation traveltime inversion. *Geophysics*, **56**:645–653.
Referenced in: 3.7, 5.2.3

Maggi, A., Tape, C., Chen, M., Chao, D., and Tromp, J. (2009). An automated time-window selection algorithm for seismic tomography. *Geophysical Journal International*, **178**(1):257–281.
Referenced in: 3.6, 4.5.5

Mai, P. M. and Thingbaijam, K. K. S. (2014). SRCMOD: An Online Database of Finite-Fault Rupture Models. *Seismological Research Letters*, **85**(6):1348–1357. doi:10.1785/0220140077.
Referenced in: B.3.5

Masson, Y., Cupillard, P., Capdeville, Y., and Romanowicz, B. (2013). On the numerical implementation of time-reversal mirrors for tomographic imaging. *Geophys. J. Int.*, **196**(3):1580–1599. doi:10.1093/gji/ggt459.
Referenced in: A.2.2

McNamara, D. E. and Buland, R. P. (2004). Ambient Noise Levels in the Continental United States. *Bulletin of the Seismological Society of America*, **94**(4):1517–1527. doi:10.1785/012003001.
Referenced in: 4.5.8

Megies, T., Beyreuther, M., Barsch, R., Krischer, L., and Wassermann, J. (2011). ObsPy – What can it do for data centers and observatories? *Annals Of Geophysics*, **54**(1):47–58.
Referenced in: 2, 2.2.5, 3.3, 4.4.2, A.3, A.6, B.2.2

Megies, T. and Wassermann, J. (2014). Microseismicity Observed at a Non-Pressure-Stimulated Geothermal Power Plant. *Geothermics*.
Referenced in: 2.5

Meier, U., Curtis, A., and Trampert, J. (2007). Global crustal thickness from neural network inversion of surface wave data. *Geophysical Journal International*, **169**(2):706–722. doi:10.1111/j.1365-246X.2007.03373.x.
Referenced in: 5.2.1

Minson, S. E. and Dreger, D. S. (2008). Stable inversions for complete moment tensors. *Geophysical Journal International*, **174**(2):585–592. doi:10.1111/j.1365-246X.2008.03797.x.
Referenced in: B.3.6, B.3.6

Modrak, R. and Tromp, J. (2016). Seismic waveform inversion best practices: regional, global, and exploration test cases. *Geophysical Journal International*, **206**(3):1864–1889. doi:10.1093/gji/ggw202.
Referenced in: 5.2.2

Mohorovičić, A. (1910). Das Beben vom 8. Oktober 1909. *Jahrbuch des meteorologischen Observatoriums in Zagreb für 1909 (translated title)*, pages 1–67.
Referenced in: 1.1

Molinari, I., Clinton, J., Kissling, E., Hetényi, G., Giardini, D., Stipčević, J., Dasović, I., Herak, M., Šipka, V., Wéber, Z., Gráczer, Z., Solarino, S., the Swiss-AlpArray Field Team, and the AlpArray Working Group (2016). Swiss-AlpArray temporary broadband seismic stations deployment and noise characterization. *Advances in Geosciences*, **43**:15–29. doi:10.5194/adgeo-43-15-2016.
Referenced in: 1.1

Montagner, J. and Kennett, B. L. N. (1996). How to reconcile body-wave and normal-mode reference earth models. *Geophys. J. Int.*, **125**:229–248. doi:10.1111/j.1365-246X.1996.tb06548.x.
Referenced in: A.2.3, A.6, B.1, B.4

Monteiller, V., Chevrot, S., Komatitsch, D., and Fuji, N. (2012). A hybrid method to compute short-period synthetic seismograms of teleseismic body waves in a 3-D regional model. *Geophys. J. Int.*, **192**(1):230–247. doi:10.1093/gji/ggs006.
Referenced in: A.2.2

MPI Forum (2009). Message Passing Interface (MPI) Forum Home Page. `http://www.mpi-forum.org/`.
Referenced in: 4.2.1

National Imagery and Mapping Agency (2000). Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems. *Technical Report TR8350.2*, National Imagery and Mapping Agency, St. Louis, MO, USA.
Referenced in: B.3.2

Nishida, K., Kawakatsu, H., Fukao, Y., and Obara, K. (2008). Background Love and Rayleigh waves simultaneously generated at the Pacific Ocean floors. *Geophys. Res. Lett.*, **35**(L16307). doi:10.1029/2008GL034753.
Referenced in: A.5.7

Nissen-Meyer, T., Dahlen, F. A., and Fournier, A. (2007a). Spherical-earth Fréchet sensitivity kernels. *Geophys. J. Int.*, **168**(3):1051–1066. doi:10.1111/j.1365-246X.2006.03123.x.
Referenced in: A.1, A.2.2, A.2.3

Nissen-Meyer, T., Fournier, A., and Dahlen, F. A. (2007b). A two-dimensional spectral-element method for computing spherical-earth seismograms - I. Moment-tensor source. *Geophys. J. Int.*, **168**(3):1067–1092. doi:10.1111/j.1365-246X.2006.03121.x.
Referenced in: A.1, A.2.1, A.2.2, A.2.3, B.2.1

Nissen-Meyer, T., Fournier, A., and Dahlen, F. A. (2008). A 2-D spectral-element method for computing spherical-earth seismograms - II. Waves in solid-fluid media. *Geophys. J. Int.*, **174**(3):873–888. doi:10.1111/j.1365-246X.2008.03813.x.
Referenced in: A.1, A.4.1, A.4.3

Nissen-Meyer, T., van Driel, M., Stähler, S. C., Hosseini, K., Hempel, S., Auer, L., Colombi, A., and Fournier, A. (2014). AxiSEM: broadband 3-D seismic wavefields in axisymmetric media. *Solid Earth*, **5**(1):425–445. doi:10.5194/se-5-425-2014.
Referenced in: A.1, A.2, A.2.1, A.4.3, B.2.1

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer, 2nd edition.
Referenced in: 5.2.2, 5.2.2

Nolet, G. (2008). *A Breviary of Seismic Tomography: Imaging the Interior of the Earth and Sun*. Cambridge University Press.
Referenced in: A.5.3

Nuber, A., Manukyan, E., and Maurer, H. (2016). Ground topography effects on near-surface elastic full waveform inversion. *Geophysical Journal International*, **207**(1):67–71. doi:10.1093/gji/ggw267.
Referenced in: 5.5.3

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Trans. AIEE*, pages 617–644. doi:10.1109/5.989875.
Referenced in: A.9

Obayashi, M., Yoshimitsu, J., Nolet, G., Fukao, Y., Shiobara, H., Sugioka, H., Miyamachi, H., and Gao, Y. (2013). Finite frequency whole mantle P wave tomography: Improvement of subducted slab images. *Geophys. Res. Lett.*, **40**:1–6.
Referenced in: 3.1

Ohmi, S. (2014). win2sac.c. `http://www1.rcep.dpri.kyoto-u.ac.jp/~ohmi/utils/src/win2sac.c`.
Referenced in: 2.4.2

Oliphant, T. E. (2007). Python for Scientific Computing. *Computing in Science & Engineering*, **9**(3):10–20.
Referenced in: 2.1

Pérez, F. and Granger, B. E. (2007). IPython: a System for Interactive Scientific Computing. *Computing in Science and Engineering*, **9**(3):21–29. doi:10.1109/MCSE.2007.53. URL `http://ipython.org`.
Referenced in: B.1

Peter, D., Komatitsch, D., Luo, Y., Martin, R., Le Goff, N., Casarotti, E., Le Loher, P., Magnoni, F., Liu, Q., Blitz, C., Nissen-Meyer, T., Basini, P., and Tromp, J. (2011). Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes. *Geophysical Journal International*, **186**(2):721–739. doi:10.1111/j.1365-246X.2011.05044.x.
Referenced in: 3.5

Pratt, R., Shin, C., and Hicks, G. (1998). Gauss-Newton and full Newton methods in frequency domain seismic waveform inversion. *Geophysical Journal International*, **133**:341–362.
Referenced in: 3.7

Rawlinson, N., Fichtner, A., Sambridge, M., and Young, M. K. (2014). Seismic Tomography and the Assessment of Uncertainty. *Advances in Geophysics*, **55**:1–76. doi:10.1016/bs.agph.2014.08.001.
Referenced in: 5.5.2

Rawlinson, N. and Sambridge, M. (2003). Seismic traveltime tomography of the crust and lithosphere. *Advances in Geophysics*, **46**:81–199.
Referenced in: 1.1, 3.1, 5.1

Rew, R. and Davis, G. (1990). NetCDF: an interface for scientific data access. *Computer Graphics and Applications, IEEE*, **10**(4):76–82. doi:10.1109/38.56302.
Referenced in: 4.2.1, A.2.1, A.3

Richter, T., Sens-Schönfelder, C., Kind, R., and Asch, G. (2014). Comprehensive observation and modeling of earthquake and temperature-related seismic velocity changes in northern Chile with passive image interferometry. *Journal of Geophysical Research : Solid Earth*, pages 4747–4765.
Referenced in: 2.5

Rickers, F., Fichtner, A., and Trampert, J. (2012). Imaging mantle plumes with instantaneous phase measurements of diffracted waves. *Geophysical Journal International*, **190**(1):650–664. doi:10.1111/j.1365-246X.2012.05515.x.
Referenced in: 5.2.3

Rickers, F., Fichtner, A., and Trampert, J. (2013). The Iceland–Jan Mayen plume system and its impact on mantle dynamics in the North Atlantic region: Evidence from full-waveform inversion. *Earth and Planetary Science Letters*, **367**:39–51. doi:10.1016/j.epsl.2013.02.022.
Referenced in: 5.3, 5.5.4

Ritsema, J., Van Heijst, H. J., and Woodhouse, J. H. (1999). Complex shear velocity structure imaged beneath Africa and Iceland. *Science*, **286**(5446):1925–1928.
Referenced in: 5, 5.2.1

Romanowicz, B. and Dziewonski, A. (1986). Toward a Federation of Broadband Seismic Networks. *EOS*, **67**(25):24–26.
Referenced in: 1.1, 5.6

Roult, G., Montagner, J.-P., Romanowicz, B., Cara, M., Rouland, D., Pillet, R., Karczewski, J.-F., Rivera, L., Stutzmann, E., and Maggi, A. (2010). The GEOSCOPE program: Progress and challenges during the past 30 years. *Seis. Res. Lett.*, **81**:427–452.
Referenced in: 3.1

Sanchez-Sesma, F. J. (2006). Retrieval of the Green's Function from Cross Correlation: The Canonical Elastic Problem. *Bull. Seismol. Soc. Am.*, **96**(3):1182–1191. doi:10.1785/0120050181.
Referenced in: A.5.7

Scheingraber, C., Hosseini, K., Barsch, R., and Sigloch, K. (2013). ObsPyLoad: A Tool for Fully Automated Retrieval of Seismological Waveform Data. *Seismol. Res. Lett.*, **84**(3):525–531. doi:10.1785/0220120103.
Referenced in: A.6

Schiemenz, A. and Igel, H. (2013). Accelerated 3-D full-waveform inversion using simultaneously encoded sources in the time domain: application to Valhall ocean-bottom cable data. *Geophysical Journal International*, **195**(3):1970–1988.
Referenced in: 2.5

Schorlemmer, D., Euchner, F., Kästli, P., Saul, J., and Group, Q. W. (2011). QuakeML: Status of the XML-based seismological data exchange format. *Annals of Geophysics*, **54**(1):59–65. doi:10.4401/ag-4874.
Referenced in: 4.2.2

Schorlemmer, D., Wyss, A., Maraini, S., Wiemer, S., and Baer, M. (2004). Orfeus Newsletter 6(2): QuakeML - An XML schema for seismology. Accessed 7-October-2015, `http://www.orfeus-eu.org/organization/Organization/Newsletter/vol6no2/quakeml.shtml`.
Referenced in: 4.2.2

SEG Technical Standards Committee (2002). *SEG Y rev 1 Data Exchange format*. Society of Exploration Geophysicists.
Referenced in: 4.3.3

Seriani, G., Priolo, E., and Pregarz, A. (1995). Modelling waves in anisotropic media by a spectral element method. *Proceedings of the 3rd International Conference on Mathematical and Numerical Aspects of Wave Propagation*, pages 289–298.
Referenced in: 5.2.1

Shiobara, H., Baba, K., Utada, H., and Fukao, Y. (2009). Ocean bottom array probes stagnant slab beneath the Philippine Sea. *EOS*, **90**:70–71.
Referenced in: 3.1

Sigloch, K., McQuarrie, N., and Nolet, G. (2008). Two-stage subduction history under North America inferred from multiple-frequency tomography. *Nature Geoscience*, **1**(7):458–462. doi:10.1038/ngeo231.
Referenced in: 5.1

Sigloch, K. and Nolet, G. (2006). Measuring finite-frequency body-wave amplitudes and traveltimes. *Geophys. J. Int.*, **167**(1):271–287. doi:10.1111/j.1365-246X.2006.03116.x.
Referenced in: A.5.3, A.18, B.5.3

Simute, S., Steptoe, H., Cobden, L., Gokhberg, A., and Fichtner, A. (2016). Full-waveform inversion of the Japanese islands region. *Journal of Geophysical Research: Solid Earth*, **121**(5):3722–3741. doi:10.1002/2016JB012802.
Referenced in: 5.3, 5.5.1

Snoke, J. A. (2009). Traveltime Tables for iasp91 and ak135. *Seismological Research Letters*, **80**(2):260–262.
Referenced in: 2.3.1

Spakman, W. (1991). Delay-time tomography of the upper mantle below Europe, the Mediterranean and Asia Minor. *Geophys. J. Int.*, **107**:309–332.
Referenced in: 1.1, 3.1, 5.1

Stähler, S. C. and Sigloch, K. (2014). Fully probabilistic seismic source inversion – Part 1: Efficient parameterisation. *Solid Earth*, **5**(2):1055–1069. doi:10.5194/se-5-1055-2014.
Referenced in: A.5.4

Stähler, S. C., Sigloch, K., and Nissen-Meyer, T. (2012). Triplicated P-wave measurements for waveform tomography of the mantle transition zone. *Solid Earth*, **3**(2):339–354. doi:10.5194/se-3-339-2012.
Referenced in: A.5.3, B.5.3

Stehly, L., Campillo, M., and Shapiro, N. M. (2006). A study of the seismic noise from its long-range correlation properties. *J. Geophys. Res.*, **111**(B10):B10306. doi:10.1029/2005JB004237.
Referenced in: A.5.7

Stephens, C. D., Chouet, B. A., Page, R. A., Lahr, J. C., and Power, J. A. (1994). Seismological aspects of the 1989-1990 eruptions at Redoubt Volcano, Alaska: the SSAM perspective. *Journal of Volcanology and Geothermal Research*, **62**(1-4):153–182.
Referenced in: 2.4.3

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S., and Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLoS Biol*, **13**(7):e1002195. doi:10.1371/journal.pbio.1002195.
Referenced in: 4.2.1

Sukhovich, A., Bonnieux, S., Hello, Y., Irisson, J.-O., Simons, F. J., and Nolet, G. (2015). Seismic monitoring in the oceans by autonomous floats. *Nature Communications*, **6**:8027. doi:10.1038/ncomms9027.
Referenced in: 5.1

Symmes, J. C. (1780–1829). *The Symmes theory of concentric spheres, demonstrating that the earth is hollow, habitable within, and widely open about the poles.*
Referenced in: 1.1

Tape, C., Liu, Q., Maggi, A., and Tromp, J. (2010). Seismic tomography of the southern California crust based on spectral-element and adjoint methods. *Geophysical Journal International*, **180**(1):433–462. doi:10.1111/j.1365-246X.2009.04429.x.
Referenced in: 1.2, 3.7, 4.5.5, 5.1, 5.2, 5.3, 5.5.1

Tape, C., Liu, Q., and Tromp, J. (2007). Finite-frequency tomography using adjoint methods - Methodology and examples using membrane surface waves. *Geophysical Journal International*. doi:10.1111/j.1365-246X.2006.03191.x.
Referenced in: 5.2.2, 5.2.3

Tarantola, A. (1988). Theoretical background for the inversion of seismic waveforms, including elasticity and attenuation. *Pure Appl. Geophys.*, **128**:365–399.
Referenced in: 1.2, 3.1, 3.1, 3.7, 5.1

Tárraga, M., Carniel, R., Ortiz, R., Marrero, J. M., and García, A. (2006). On the predictability of volcano-tectonic events by low frequency seismic noise analysis at Teide-Pico Viejo volcanic complex, Canary Islands. *Natural Hazards and Earth System Science*, **6**(3):365–376.
Referenced in: 2.4.3

The HDF Group (1997–2015). Hierarchical Data Format, version 5. `https://www.hdfgroup.org/HDF5/`.
Referenced in: 4.2.1

The International Federation of Digital Seismograph Networks (FDSN) (2014). FDSN StationXML Schema. `http://www.fdsn.org/xml/station/`.
Referenced in: 2.3.2

Thompson, D. A. and Best, J. S. (2000). The future of magnetic data storage techology. *IBM Journal of Research and Development*, **44**(3):311–322. doi:10.1147/rd.443.0311.
Referenced in: 4.1

Thomson, D. (1982). Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, volume 70, pages 1055–1096.
Referenced in: 5.2.3

Tolman, H. (2009). User manual and system documentation of WAVEWATCH-III version 3.14. *Technical Report 276*.
Referenced in: A.22

Tong, P., Komatitsch, D., Tseng, T.-L., Hung, S.-H., Chen, C.-W., Basini, P., and Liu, Q. (2014). A 3-D spectral-element and frequency-wave number hybrid method for high-resolution seismic array imaging. *Geophysical Research Letters*, **41**:7025–7034.
Referenced in: 3.11

Trabant, C., Hutko, a. R., Bahavar, M., Karstens, R., Ahern, T., and Aster, R. (2012). Data Products at the IRIS DMC: Stepping Stones for Research and Other Applications. *Seismological Research Letters*, **83**(5):846–854. doi:10.1785/0220120032.
Referenced in: B.5.5

Tromp, J. (2007). Theory and Observations – Forward Modeling and Synthetic Seismograms: 3-D Numerical Methods. *Treatise on Geophysics*.
Referenced in: A.1

Tromp, J., Komatitsch, D., Hjörleifsdóttir, V., Liu, Q., Zhu, H., Peter, D., Bozdağ, E., McRitchie, D., Friberg, P., Trabant, C., and Hutko, A. (2010). Near real-time simulations of global CMT earthquakes. *Geophysical Journal International*, **183**:381–389. doi:10.1111/j.1365-246X.2010.04734.x.
Referenced in: 4.5.4, A.1, A.1, A.5.2, B.6

Tromp, J., Komattisch, D., and Liu, Q. (2008). Spectral-Element and Adjoint Methods in Seismology. *Communications in Computational Physics*, **3**(1):1–32.
Referenced in: 2.3.1

Tromp, J., Tape, C., and Liu, Q. (2005). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophysical Journal International*, **160**:195–216.
Referenced in: 1.2, 3.1, 3.7, 4.5.5, 5.1, 5.2.2, 5.3

Tsai, V. C. (2009). On establishing the accuracy of noise tomography travel-time measurements in a realistic medium. *Geophys. J. Int.*, **178**(3):1555–1564. doi:10.1111/j.1365-246X.2009.04239.x.
Referenced in: A.5.7

Valentine, A. P. and Woodhouse, J. H. (2010). Reducing errors in seismic tomography: combined inversion for sources and structure. *Geophys. J. Int.*, **180**(2):847–857. doi:10.1111/j.1365-246X.2009.04452.x.
Referenced in: A.5.4

van der Hilst, R. D., Kennett, B. L. N., Christie, D., and Grant, J. (1994). Project SKIPPY explores the lithosphere and mantle beneath Australia. *EOS*, **75**:180–181.
Referenced in: 3.1

van Driel, M., Krischer, L., Stähler, S. C., Hosseini, K., and Nissen-Meyer, T. (2015). Instaseis: instant global seismograms based on a broadband waveform database. *Solid Earth*, **6**(2):701–717. doi:10.5194/se-6-701-2015.
Referenced in: B.2, B.2.1

van Driel, M. and Nissen-Meyer, T. (2014a). Optimized viscoelastic wave propagation for weakly dissipative media. *Geophys. J. Int.*, **199**(2):1078–1093. doi:10.1093/gji/ggu314.
Referenced in: A.1, A.4.1, B.2.1

van Driel, M. and Nissen-Meyer, T. (2014b). Seismic wave propagation in fully anisotropic axisymmetric media. *Geophys. J. Int.*, **199**(2):880–893. doi:10.1093/gji/ggu269.
Referenced in: A.1, A.4.1, B.2.1

Waldhauser, F. and Ellsworth, W. L. (2000). A Double-Difference Earthquake Location Algorithm: Method and Application to the Northern Hayward Fault, California. *Bulletin of the Seismological Society of America*, **90**(6):1353–1368. doi:10.1785/0120000006.
Referenced in: B.5.2

winformat (2014). manpage of winformat. `http://eoc.eri.u-tokyo.ac.jp/cgi-bin/show_man_en?winformat`.
Referenced in: 2.4.2

Yomogida, K. (1992). Fresnel zone inversion for lateral heterogeneities in the Earth. *Pure Appl. Geophys.*, **138**:391–406.
Referenced in: 3.1, 5.1

Yoshizawa, K. and Kennett, B. L. N. (2004). Multi-mode surface wave tomography for the Australian region using a 3-stage approach incorporating finite-frequency effects. *J. Geophys. Res.*, **109**:doi:10.1029/2002JB002254.
Referenced in: 3.1, 5.1

Yoshizawa, K. and Kennett, B. L. N. (2005). Sensitivity kernels for finite-frequency surface waves. *Geophys. J. Int.*, **162**:910–926.
Referenced in: 3.1, 5.1

Yuan, Y. O., Simons, F. J., and Tromp, J. (2016). Double-difference adjoint seismic tomography. *Geophysical Journal International*, **206**(3).
Referenced in: 5.6

Zhou, Y., Dahlen, F. A., and Nolet, G. (2004). Three-dimensional sensitivity kernels for surface wave observables. *Geophys. J. Int.*, **158**:142–168.
Referenced in: 3.7

Zhu, H., Bozdağ, E., Peter, D., and Tromp, J. (2012). Structure of the European upper mantle revealed by adjoint tomography. *Nature Geoscience*, **5**(7):493–498. doi:10.1038/ngeo1501.
Referenced in: 1.2, 3.1, 4.5.5, 5.1, 5.3

Zoeppritz, K. B. (1907). Über Erdbebenwellen II. Laufzeitkurven. *Nachrichten der Königlichen Gesellschaft der Wissenschaften zu Göttingen. Mathematisch-Physikalische Klasse*, pages 529–549.
Referenced in: 1.1

Zoeppritz, K. B., Geiger, L., and Gutenberg, B. (1912). Über Erdbebenwellen V. Konstitution des Erdinnern, erschlossen aus dem Bodenverrückungsverhalten der einmal reflektierten zu den direkten longitudinalen Erdbebenwellen, und einige andere Beobachtungen über Erdbebenwellen. *Nachrichten der Königlichen Gesellschaft der Wissenschaften zu Göttingen. Mathematisch-Physikalische Klasse*, pages 121–206.
Referenced in: 1.1

A
Instaseis

While not a core part of this disseration, the Instaseis software was created during the course of this thesis.

It is a new method and implementation to store global Green's functions in a database which allows for near-instantaneous (on the order of milliseconds) extraction of arbitrary seismograms. Using the axisymmetric spectral element method (AxiSEM), the generation of these databases, based on reciprocity of the Green's functions, is very efficient and is approximately half as expensive as a single AxiSEM forward run. Thus, this enables the computation of full databases at half the cost of the computation of seismograms for a single source in the previous scheme and allows to compute databases at the highest frequencies globally observed. By storing the basis coefficients of the numerical scheme (Lagrange polynomials), the Green's functions are 4th order accurate in space and the spatial discretization respects discontinuities in the velocity model exactly. High order temporal interpolation using Lanczos resampling allows to retrieve seismograms at any sampling rate. AxiSEM is easily adaptable to arbitrary spherically symmetric models of Earth as well as other planets.

This chapter presents the basic rationale and details of the method as well as benchmarks and illustrate a variety of applications. The code is open source and available with extensive documentation at `www.instaseis.net` (last accessed March 2017). This chapter has been published in:

van Driel, M., Krischer, L., Stähler, S., Hosseini, K., & Nissen-Meyer, T. (2015).
Instaseis: instant global seismograms based on a broadband waveform database.
Solid Earth, 6(2), 701–717.
`https://doi.org/10.5194/se-6-701-2015`

A.1 Introduction

Despite the exponential growth of computational power and substantial progress of 3-D numerical methods for seismic wave propagation in the last 15 years (Igel et al., 2000; Komatitsch and Tromp, 2002b; Tromp, 2007; Tromp et al., 2010), the simulation of the highest frequencies observed in seismic waves on the global scale remains a high-performance computing challenge and is not yet done routinely. This is why many seismologists still rely on approximate methods to compute and analyze high-frequency body waves such as ray-theoretical traveltimes (e.g. the TauP-toolkit described in Crotwell et al., 1999), WKBJ synthetics (Chapman, 1978), the reflectivity method (Fuchs and Müller, 1971) or the frequency-wavenumber integration method (Kikuchi and Kanamori, 1982). More recently, several methods that include the full physics in solving the seismic wave equation while reaching the highest observable frequencies by assuming spherically symmetric models have become available, see Fig. A.1 for an example. These methods include the direct solution method (DSM, Geller and Ohminato, 1994; Kawai et al., 2006), the frequency domain integration method (GEMINI, Friederich and Dalkolmo, 1995) and a generalization of it including self gravitation (Yspec, Al-Attar and Woodhouse, 2008).

As detailed by Nissen-Meyer et al. (2007b), the main drawback of these methods when applied to computing wavefields rather than single seismograms, is their scaling proportional to the number of points in space where the wavefield is sampled. This motivated the development of a direct time-domain approach, where the displacement as a function of space and time is a natural field variable and only needs to be written to disk (Nissen-Meyer et al., 2007a,b, 2008). The implementation of this axisymmetric spectral element method AxiSEM was recently extended to include anisotropy and attenuation (van Driel and Nissen-Meyer, 2014b,a), published under public license (Nissen-Meyer et al., 2014) and is available at `www.axisem.info`.

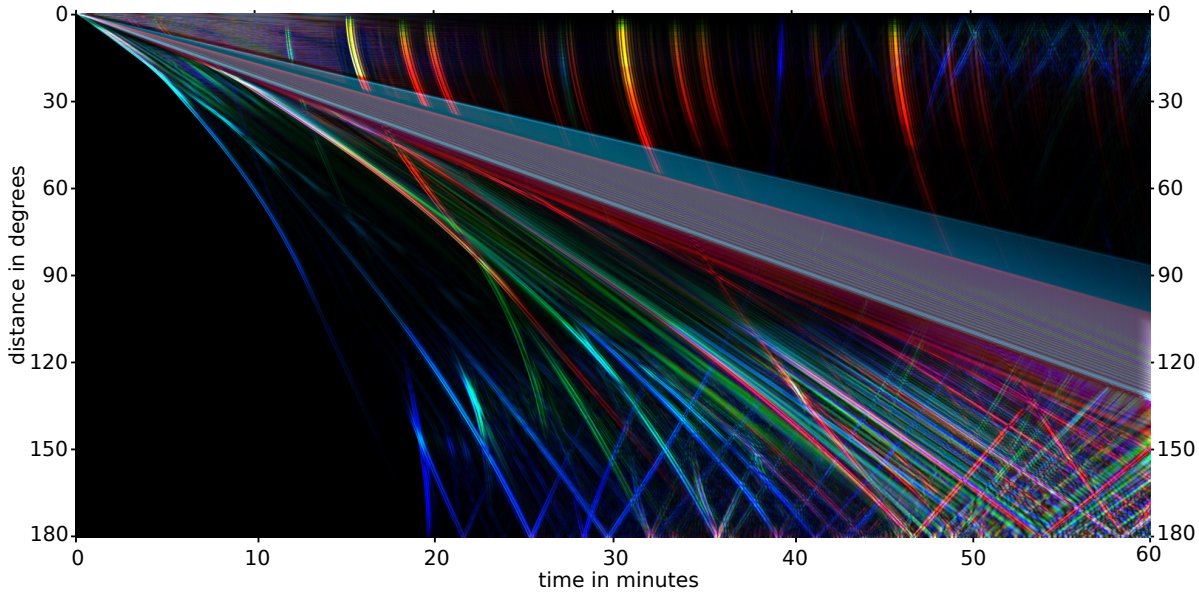

Figure A.1.: Global stack of 1h of seismograms accurate to a shortest period of 2 s for an earthquake in 27 km depth computed with Instaseis. The displacement is color-coded analogous to the IRIS global stack (Astiz et al., 1996), i.e. red = transversal component, green = radial component, blue = vertical component. An automatic gain control (AGC) with a window of 100 s length is used to balance large amplitude variations between the various phases. Note that creating this plot does not require to define the source depth at the time of database calculation.

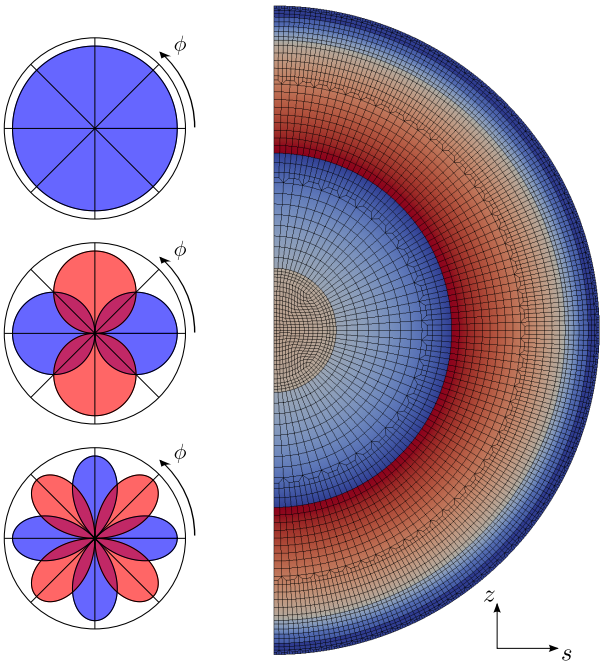

Figure A.2: The 3-D wavefield is decomposed analytically into monopole, dipole and quadrupole radiation patterns (left) and the remaining 2-D problem is solved on a D-shaped domain (right) using the spectral element method. While the forward databases require a total of four 2-D computations, it is only two for the backward databases using reciprocity of the Green's function: one for the vertical and one for the horizontal components. (modified from Nissen-Meyer et al., 2014)

As computing full global waveforms especially at higher frequencies requires substantial computational resources, several initiatives serve to deliver waveforms by means of databases without having to run a full numerical solver. The ShakeMovie project (Tromp et al., 2010) provides synthetics for earthquakes from the CMT catalogue (`www.globalcmt.org`) recorded at permanent GSN and FDSN stations in 1-D and 3-D velocity models. The Pyrocko toolbox (`http://emolch.github.io/pyrocko`) provides a Python interface to generate and access Green's function databases, which for the global case are based on GEMINI, several databases are offered for download. In this chapter we present a method that uses AxiSEM to generate global Green's function databases and provides a Python interface for convenient extraction of seismograms. The advantage over ShakeMovie synthetics are the possible higher frequencies and arbitrary source and receiver combinations independent of catalogues and real stations. Compared to Pyrocko with GEMINI synthetics, AxiSEM is more efficient in generating the databases, allowing to routinely compute them for a large number of different background models or specialized applications (e.g. limited depth/distance ranges). Also, by using the Lagrangian polynomials in the SEM mesh as basis functions, it achieves higher spatial accuracy.

This chapter is structured as follows. In section A.2 we present the technical aspects and argue for the choices made in the spatial and temporal discretization. Section A.3 gives a short overview of the Python interface. In section A.4 we show the performance with respect to accuracy, speed and disk space requirements for the databases. Finally, we depict a variety of applications in section A.5.

A.2 Methods

A.2.1 Computing Green's Functions with AxiSEM

AxiSEM was designed from the beginning with the application of computing global wavefields rather then single seismograms in mind (Nissen-Meyer et al., 2007b). This becomes apparent

in the following main advantages in this application: It uses a 2-D discretization (Fig. A.2), with an analytical decomposition of the 3-D wavefield into several 2-D wavefields. For moment tensor sources, four 2-D wavefields are needed, for force sources, two. As it is a time domain method, the displacement field in space-time is a natural field variable of the numerical scheme and simply needs to be written to disk without any extra computational cost when larger regions of Earth are included in the database. AxiSEM uses a spectral element scheme for spatial discretization which lends itself well to parallelization on HPC systems. As it is based on the weak formulation of the wave equation, it naturally includes the free surface boundary condition and allows for highly accurate modelling of surface waves.

Nissen-Meyer et al. (2014) argued against using collective parallel I/O since the availability of the NetCDF libraries (Rew and Davis, 1990) was not granted on all supercomputers. For that reason, we implemented a round robin I/O scheme, which remains advantageous when running AxiSEM on less than about 100 cores in parallel and to avoid installation problems on systems where NetCDF is not available as a pre-compiled package. On supercomputers however, the situation has since improved and NetCDF compiled with parallel support seems now to be widespread. For this reason, we implemented a collective parallel I/O scheme that performs well, even when running on more than 1000 cores, see Table A.1. In this scheme, all processes that have to write data to disk communicate via the message passing interface (MPI) and then write collectively at the same time to the parallel file system. This way we achieved throughputs of up to 4 GB/s on SuperMUC.

A.2.2 Forward and Backward Databases

Instaseis has the capability of dealing with forward wavefields, i.e. the waves are propagated from a moment-tensor point source at fixed depth (i.e. receivers exist throughout the medium), as well as backward or reciprocal wavefields, where the wavefields are propagated from a single-force point source at fixed depth and recorded throughout the medium (i.e. sources exist throughout the medium).

Potential applications of forward databases are the generation of 3-D wave-propagation movies (Holtzman et al., 2013), the computation of incoming teleseismic waves in 1-D/3-D hybrid methods (e.g. Monteiller et al., 2012; Masson et al., 2013) or the forward field in the computation of sensitivity kernels (Nissen-Meyer et al., 2007a) for seismic tomography. To generate a forward database, a total of four runs with AxiSEM are needed (Nissen-Meyer et al., 2007b).

In contrast, reciprocal databases utilize the reciprocity of the Green's functions, and are useful in all cases where the receivers are at fixed depth, thus for instance mimicking earthquake catalogues recorded at stations along the surface. The source can be located anywhere in the region where the Green's functions are recorded in the simulation, thus allowing for unlimited choices in the source-receiver geometry. To generate a reciprocal database, a total of two runs with AxiSEM are needed, one for the vertical component and one for both horizontal components of the seismogram (Nissen-Meyer et al., 2007b). It is also possible to compute a database for the vertical component seismograms only, which is then a factor of three faster and uses only about 40% of the disk space.

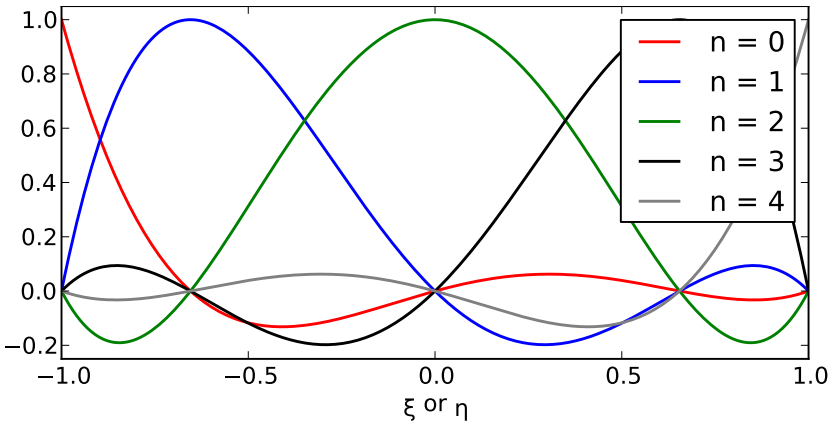

Figure A.3.: Lagrangian basis polynomials $l_n(\xi)$ of fourth order in one dimension. At the collocation points, all but one are zero, such that the value of the interpolated function at this point coincides with the coefficient in this basis expansion.

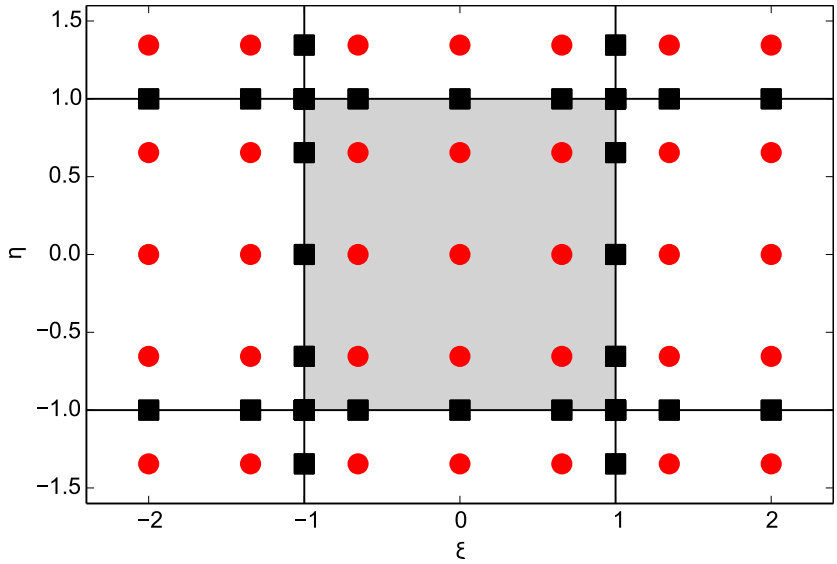

Figure A.4.: Lagrange interpolation points inside an element (gray) and its neighbours. Coordinates ξ and η are the reference coordinates of the gray element. Points on the edges (black squares) are shared between neighbours and function values at these points need to be stored only once if the function is continuous (e.g. displacement). The number of global degrees of freedom per element of such functions is thus approximately 16 compared to 25 for discontinuous functions (e.g. strain).

A.2.3 The Spatial Scheme

For the spatial discretization we choose to keep the same basis as used in AxiSEM. The displacement $\vec{u}$ within each element is expanded in terms of Lagrangian polynomials l_i (see Fig. A.3) of order N defined on the integration points of the spectral element scheme (see Fig. A.4):

$$\vec{u}(\xi, \eta, t) = \sum_{ij=0}^{N} \vec{u}_{ij}(t) l_i(\xi) l_j(\eta) \tag{A.1}$$

ξ and η are the reference coordinates of the element and N typically has a value of 4. This approach has several advantages:

- The wavefield is represented by polynomials, typically of degree 4, interpolation is hence of 4th order accuracy.
- The basis is local and only few coefficients are needed to represent the wavefield inside an element (typically 25), in contrast to e.g. global basis functions such as spherical harmonics.
- Discontinuities in the model that cause discontinuities in the strain Green's functions are respected by the mesh.
- The strain tensor (representing the moment tensor in the reciprocal case) can be computed on the fly from the stored displacements at high accuracy. This reduces the storage by a factor of two as the displacement has three degrees of freedom, compared to six for the strain.
- Since the displacement is continuous also at model discontinuities and element boundaries, it needs to be stored only once at all Gauss-Lobatto-Legendre (GLL) points that belong to multiple elements, reducing the storage by another factor of $16/25 = 0.64$ (see Fig. A.4).
- Storing the displacement allows to use force sources as well without any extra computation or storage requirements.

Fig. A.5 visualizes the spatial representation for a long period mesh (50s) for the Rayleigh wave train and the $G_{rr,r}$ component of the strain Green's tensor: the strain is smooth also across the doubling layer of the mesh where the background model (ak135f, Montagner and Kennett, 1996) is smooth as well. Still, the discontinuities of the model and hence the strain are explicitly represented by this discretization and the resolution of the mesh is adapted to the local wavelength, as for instance in the crust. Figure A.6 shows an example for 2 s shortest period and compares the SEM discretization to regular depth sampling. In the regular sampling case with nearest neighbour interpolation, the phase and envelope errors can be quite large, especially close to the model discontinuities (up to 80% envelope misfit and 4% phase misfit as defined by Kristekova et al. (2009)) and for very shallow sources (up to 40% envelope misfit and 14% phase misfit).

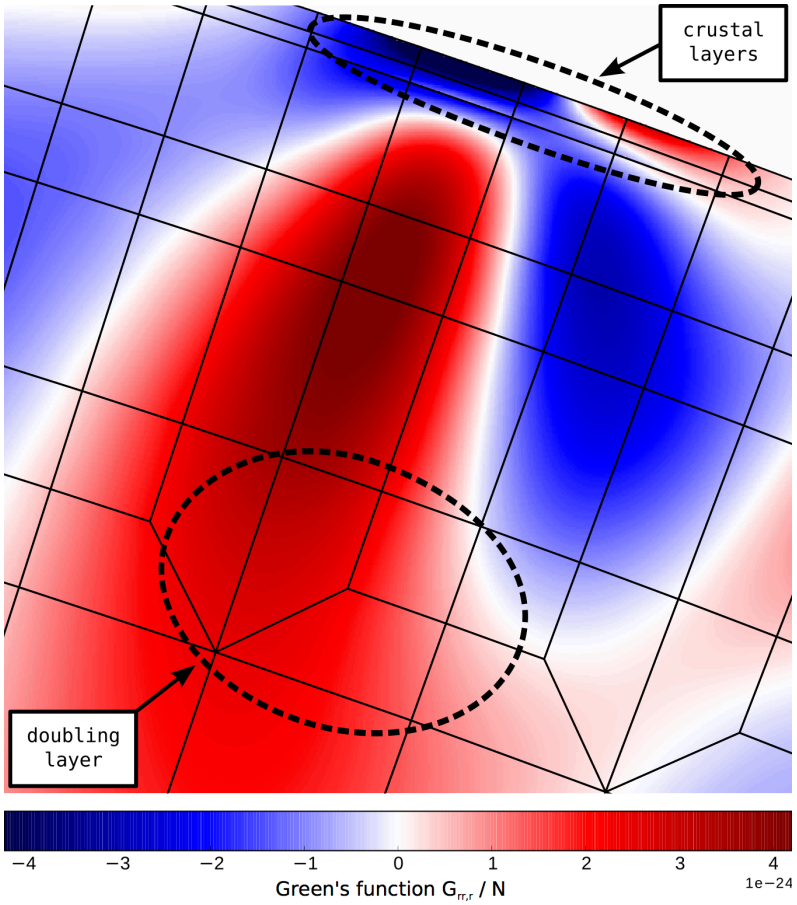

Figure A.5: Snapshot of one component of the Green's tensor ($G_{rr,r}$) as represented in the SEM basis for a shortest period of 50s. Discontinuities such as caused by the crustal layers are exactly represented and the wavefield is smooth across doubling layers of the mesh.

Finite Element Mapping

One performance-critical step in the spatial scheme is to find the reference coordinates (ξ, η) inside the spectral element that includes a point given in global coordinates (s, z). While the opposite mapping is trivial because this is how the elements of the SEM are defined (Nissen-Meyer et al., 2007a, appendix A1), it cannot be generally inverted easily. Hua (1990) presents an analytical inverse solution for quadrilateral elements, which is quite involved and not easy to generalize to the semicircular elements used in AxiSEM.

We follow a two-step approach to finding the reference coordinates. First, we find the six closest element midpoints to limit the search to a small number of candidate elements in which the point could be. The number six is specific to the AxiSEM mesh, where each corner point can belong to a maximum of six elements in the doubling layers, see Fig. A.7. This step can be seen as approximating the AxiSEM mesh with Voronoi cells. For most points, the closest midpoint will already indicate the correct element, in the worst case the second step has to be performed for all six candidates.

In a second step, the reference coordinates (ξ, η) of the given point (s_p, z_p) are computed for the six candidate elements sorted by the distance of the midpoints. If both ξ and η are in the interval [-1, 1], the element is found. (ξ, η) are computed using an iterative gradient scheme

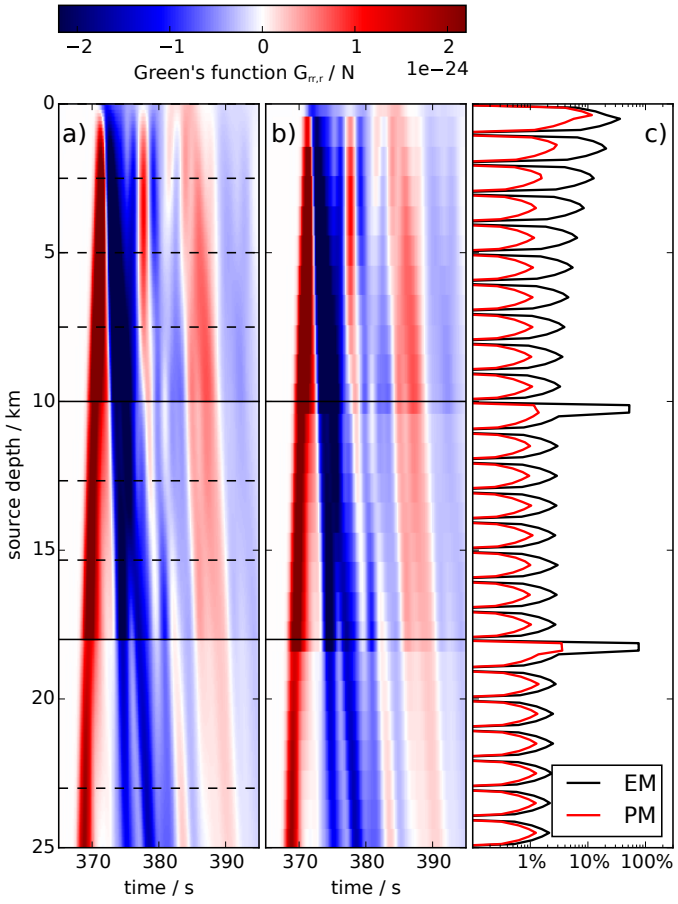

Figure A.6: One component of the strain Green's tensor ($G_{rr,r}$) for a distance of 30° as a function of time and depth with a shortest period of 2 s. (a) SEM basis vs. (b) regular sampling with 1 km distance and (c) phase and envelope misfits (EM and PM in the legend, see Kristekova et al., 2009) caused by the regular sampling computed in the period range 1-20 s. Dashed lines in the left panel sketch the spectral elements. The crustal discontinuities of ak135f (Montagner and Kennett, 1996) are indicated by solid lines and lead to discontinuities in $G_{rr,r}$, which are exactly represented in the SEM basis.

adopted from SPECFEM3D (Komatitsch and Tromp, 2002a). Starting from the midpoint of the candidate element, updated values are found by linear approximation of the inverse mapping:

$$\begin{pmatrix} \xi_{n+1} \\ \eta_{n+1} \end{pmatrix} = \begin{pmatrix} \xi_n \\ \eta_n \end{pmatrix} + \boldsymbol{\mathcal{J}}^{-1}(\xi_n, \eta_n) \cdot \begin{pmatrix} s_p - s(\xi_n, \eta_n) \\ z_p - z(\xi_n, \eta_n) \end{pmatrix} \tag{A.2}$$

with the Jacobian matrix defined as

$$\boldsymbol{\mathcal{J}}(\xi, \eta) = \begin{pmatrix} \partial_\xi s & \partial_\eta s \\ \partial_\xi z & \partial_\eta z \end{pmatrix}, \tag{A.3}$$

and the mapping $s(\xi, \eta)$ and $z(\xi, \eta)$ depending on the element type as defined in Nissen-Meyer et al. (2007b). In the AxiSEM mesh, this iteration converges to numerical accuracy within less than ten iterations and is not performance critical for Instaseis as it is only used on the few candidate elements. Also, this two step approach requires only the midpoints of all elements in the mesh to be read from file on initialization and can be implemented efficiently using the kd-tree provided by the SciPy package (Jones et al., 2001).

A.2.4 The Temporal Scheme

The design of the temporal scheme is guided by a number of constraints on the spectrum of the source time function: the spectrum should decay steep enough above the highest frequency resolved by the mesh, such that the least number of samples according to the Nyquist criterion

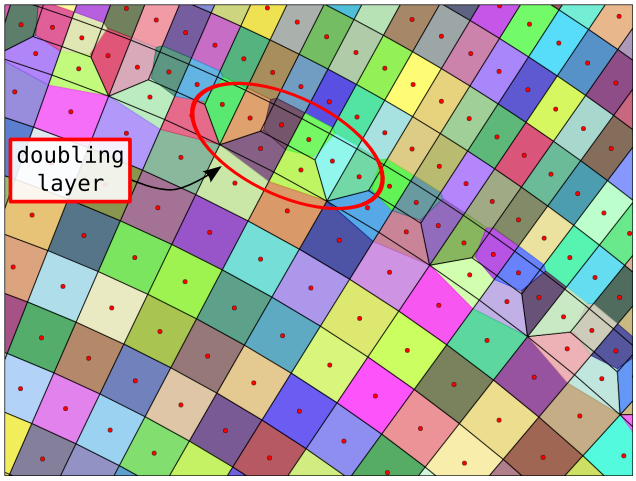

Figure A.7: Voronoi approximation (colored) of the AxiSEM mesh (black lines) using the midpoints of the elements (red circles) only, zoomed onto a doubling layer for a 50 s mesh. For most elements, the Voronoi cell coincides almost exactly with the AxiSEM element, note that most of the AxiSEM elements have edges of concentric circles while the edges of the Voronoi cells are all straight lines. In the worst case six AxiSEM elements have to be tested whether a point is inside or not.

can be used without introducing aliasing. On the other hand, it should not decay too steeply, such that it is still possible to deconvolve and convolve with another source time function. Additionally, the spectrum should be as flat as possible within the usable frequency range and 'earthquake-like' without the necessity of deconvolving it when extracting a seismogram from the database. An actual delta function as would be required for true Green's functions cannot be represented in a discrete approximation as it is not bandlimited.

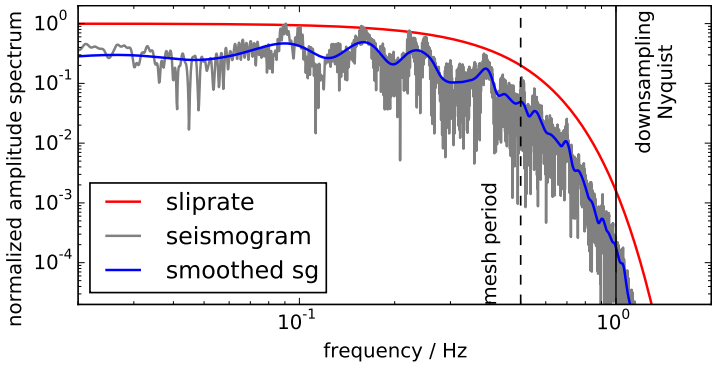

Figure A.8: Normalized amplitude spectra of the Gaussian source time function (sliprate) used at 2 s mesh period and a vertical component synthetic seismogram recorded at 40° epicentral distance. The vertical lines denote the resolution of the mesh and the Nyquist frequency of the downsampling using 4 samples per mesh period.

We found a Gaussian source time function with $\sigma = \tau/3.5$ to fulfill these requirements, where τ is the shortest period resolved by the mesh. Fig. A.8 shows the amplitude spectra of this source time function as well as a corresponding velocity seismogram at a distance of 40°. The two spectra have a very similar general shape and decay to 10^{-3} of the maximum at half the shortest period. This motivates that sampling with four samples per period will not introduce aliasing artifacts.

It is desirable to retrieve seismograms from the database with arbitrary time steps, which requires interpolation or resampling. Popular time domain schemes such as interpolation by low order polynomials or splines do not work well close to the Nyquist frequency. On the other hand, frequency domain resampling by zero-padding the discrete Fourier transform of the signal can only resample to rational multiples of the original sampling interval. Finally, the kernel from the theoretically exact reconstruction according to the Nyquist-Shannon sampling theorem (i.e. the sinc function) has infinite support which renders it impractical as well (see Burger and Burge, 2009, section 10.3 for an extended introduction to interpolation).

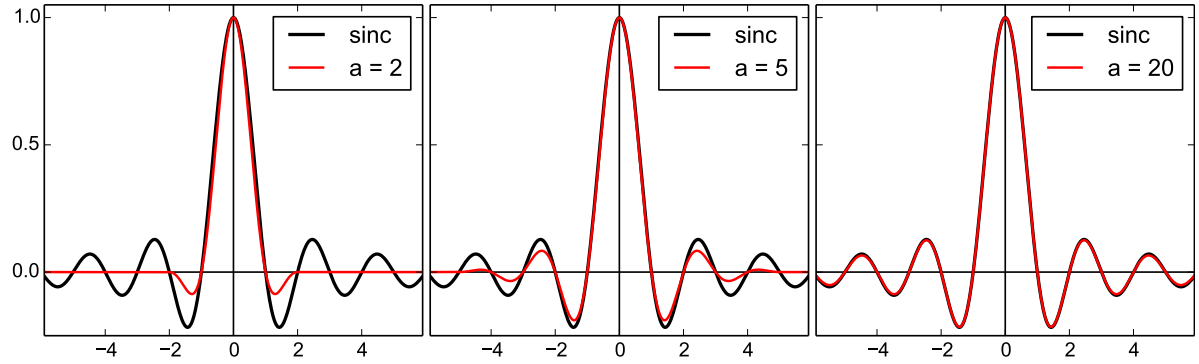

Figure A.9.: Lanczos kernels used for resampling. For large values of the parameter a, it converges towards the $sinc$ function, which is the kernel that allows exact reconstruction for bandlimited signals as stated in the Nyquist sampling theorem (Nyquist, 1928).

Therefore, we adopt the Lanczos resampling scheme, which is popular in image processing and an approximation to the sinc-resampling with finite support. The Lanczos kernel is defined as the sinc function multiplied by the Lanczos window function (Burger and Burge, 2009):

$$L(t) = \begin{cases} sinc(t)\, sinc(t/a) & \text{if } t \in [-a, a] \\ 0 & \text{else,} \end{cases} \tag{A.4}$$

where a is a parameter to control the number of samples to be used in the interpolation and the sinc function is defined as

$$sinc(t) = \frac{\sin(\pi t)}{\pi t}. \tag{A.5}$$

Interpolation is then performed by convolving the discrete signal s_i with this kernel and evaluating it at the new timesamples t_j (Burger and Burge, 2009):

$$S(t_j) = \sum_{i=\lfloor x \rfloor - a + 1}^{\lfloor x \rfloor + a} s_i L(t_j - i), \tag{A.6}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Figure A.9 shows the Lanczos kernel for different values of a, Fig. A.10 shows a practical example of resampling a seismogram. In Fig. A.11 we test a number of values for a for the first 1800 s of the same seismogram and we find $a = 12$ to be a reasonable compromise between cost (using 25 samples in the interpolation) and accuracy (RMS error of 0.03%).

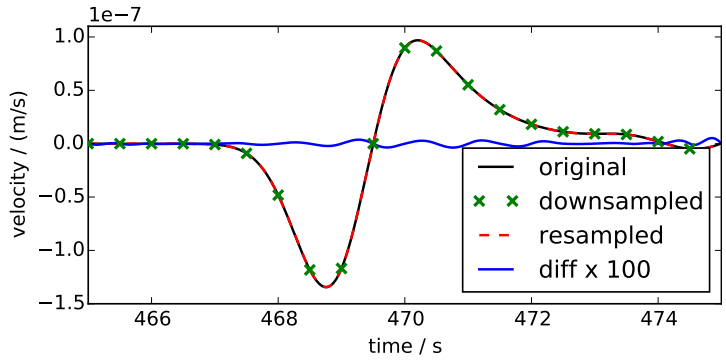

Figure A.10: Resampling using a Lanczos kernel with $a = 12$ of a P arrival velocity seismogram convolved with the source time function from Fig. A.8 recorded at 40°distance, and the resampling error multiplied by a factor of 100. The relative error is on the order of 0.05% of RMS, see Fig. A.11.

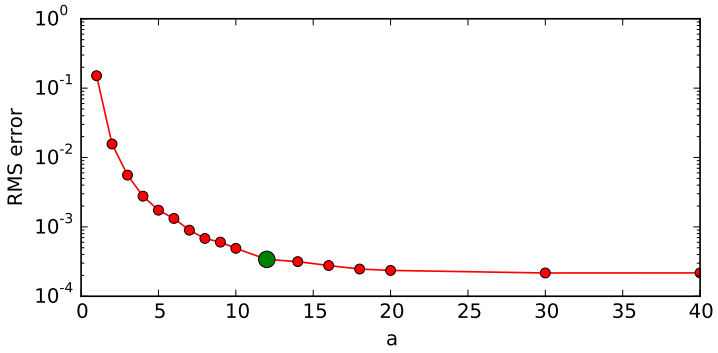

Figure A.11: RMS error of the resampling using the first 1800s of the seismogram from Fig.A.10. Convergence is reached around $a = 20$. It does not converge to zero, because some high frequency energy was negelected in the downsampling, see Fig. A.8.

A.3 Python API

Instaseis is implemented as a library for the Python programming language with some performance critical parts written in Fortran. Furthermore it directly integrates with the ObsPy package (Megies et al., 2011; Beyreuther et al., 2010) and utilizes the Python bindings to NetCDF 4 (Rew and Davis, 1990). This enables it to take advantage of the strong scientific Python ecosystem built on top of the SciPy Stack (Jones et al., 2001). Reasons for choosing Python include its growing popularity in the sciences and it being easy to learn and use while still sufficiently powerful for complex scenarios. Python is open-source and particularly well suited for big data applications and the integration with web services and databases which suits the potential uses for Instaseis.

```
>>> import instaseis
>>> db = instaseis.open_db("./AK135")
>>> receiver = instaseis.Receiver(network="BW", station="ZUGS",
                                  latitude=47.416, longitude=10.979)
>>> source = instaseis.Source(
        latitude=89.91, longitude=0.0, depth_in_m=12000,
        m_rr = 4.71e+17, m_tt = 3.81e+15, m_pp =-4.74e+17,
        m_rt = 3.99e+16, m_rp =-8.05e+16, m_tp =-1.23e+17)
>>> st = db.get_seismograms(source=source, receiver=receiver)
>>> print(st)
3 Trace(s) in Stream:
BW.ZUGS..LXZ | 1970-01-01T00:00:00.00Z - ... | 2.1 Hz, 7746 samples
BW.ZUGS..LXN | 1970-01-01T00:00:00.00Z - ... | 2.1 Hz, 7746 samples
BW.ZUGS..LXE | 1970-01-01T00:00:00.00Z - ... | 2.1 Hz, 7746 samples
```

Listing A.1.: The Instaseis Python API demonstrated in a short interactive Python session. A `Source` and a `Receiver` object are created and then passed to the `get_seismograms()` method of an `InstaseisDB` object. This will extract the Green's functions from the databases and perform all necessary subsequent steps resulting in directly usable three-component seismograms in form of an ObsPy `Stream` object. Please refer to the Instaseis documentation for details.

Listing A.1 shows how to use the Python API in the most simple case. Instaseis provides an object oriented interface: in addition to the shown `Source` and `Receiver` classes it furthermore provides `ForceSource` and `FiniteSource` objects. These can also be created by providing data in most commonly used file formats like StationXML, QuakeML, and the Standard Rupture Format. Please refer to the Instaseis documentation for further details (`www.instaseis.net`).

Combining and integrating these features enables the construction of modern and clean workflows to solve new problems. A big advantage of this approach is that no temporary files need to be created and the synthetic seismograms can be extracted from the databases on demand when and where they are needed.

The Python API furthermore implements a client/server approach for remote Instaseis database access over HTTP. This enables organizations to host high-frequency databases and serve them to users over the internet. This eliminates the need and upfront cost to calculate, store, and distribute Instaseis databases for most users while still offering enough performance for many use cases. The Python interface is data-source independent: from a usage perspective it does not matter if the databases are available locally or via the internet.

Instaseis is developed with a test-driven approach utilizing continuous integration, i.e. every change in the code is automatically tested for a number of different python version once commited to the repository. It is well documented, has a high test coverage, and we intend to maintain it for the next couple of years providing a solid foundation for future applications built on top of it. It is licensed under the Lesser GNU General Public License v3.0, the source code and issue tracker are hosted on GitHub.

A.4 Benchmarks

A.4.1 Accuracy

As we already provided some rigorous validation comparing AxiSEM synthetics to a reference solution (Yspec Al-Attar and Woodhouse, 2008) in van Driel and Nissen-Meyer (2014b,a), the purpose of this section is only to confirm that using the new scheme with reciprocal computations, spatial interpolation and temporal resampling does not decrease accuracy. Fig. A.12 shows a record section and some details for Instaseis, AxiSEM and Yspec seismograms computed in the anisotropic, visco-elastic PREM model for an event at 126 km depth beneath Tonga bandpass filtered to 50 s to 2 s period.

While this figure is similar and the AxiSEM and Yspec reference data actually the same as presented in van Driel and Nissen-Meyer (2014a, Fig. 11), it is important to note that they were generated in very different ways: here we computed a whole Green's function database for all epicentral distances and down to 700 km source depth and changing source or receivers would cost a few milliseconds only. In our previous approach, this would have required a full new AxiSEM simulation on the order of 10K CPU hours computational cost. Also, in contrast to van Driel and Nissen-Meyer (2014a), we used default mesh parameters for 2 s period and time step close to the stability limit of the 4th order symplectic time scheme (Nissen-Meyer et al., 2008). Still, the phase misfit (Kristekova et al., 2009) is well below 1% in all zoom windows and the maximum of the envelope misfit is 2% for the PPP phase on station ALE.

The fact that these traces are virtually indistinguishable for such a demanding setup of wave propagation over 800 wavelengths (waves at 2 s period travelling for 1600 s) verifies that the entire workflow of computing and querying the database are correctly implemented. In particular, numerical reciprocity (i.e. the different force and moment sources), on-the-fly calculation of the strain tensor as well as temporal and spatial sampling have no significant adverse effect on accuracy, i.e. any remaining errors vanish within numerical accuracy of the forward solver AxiSEM.

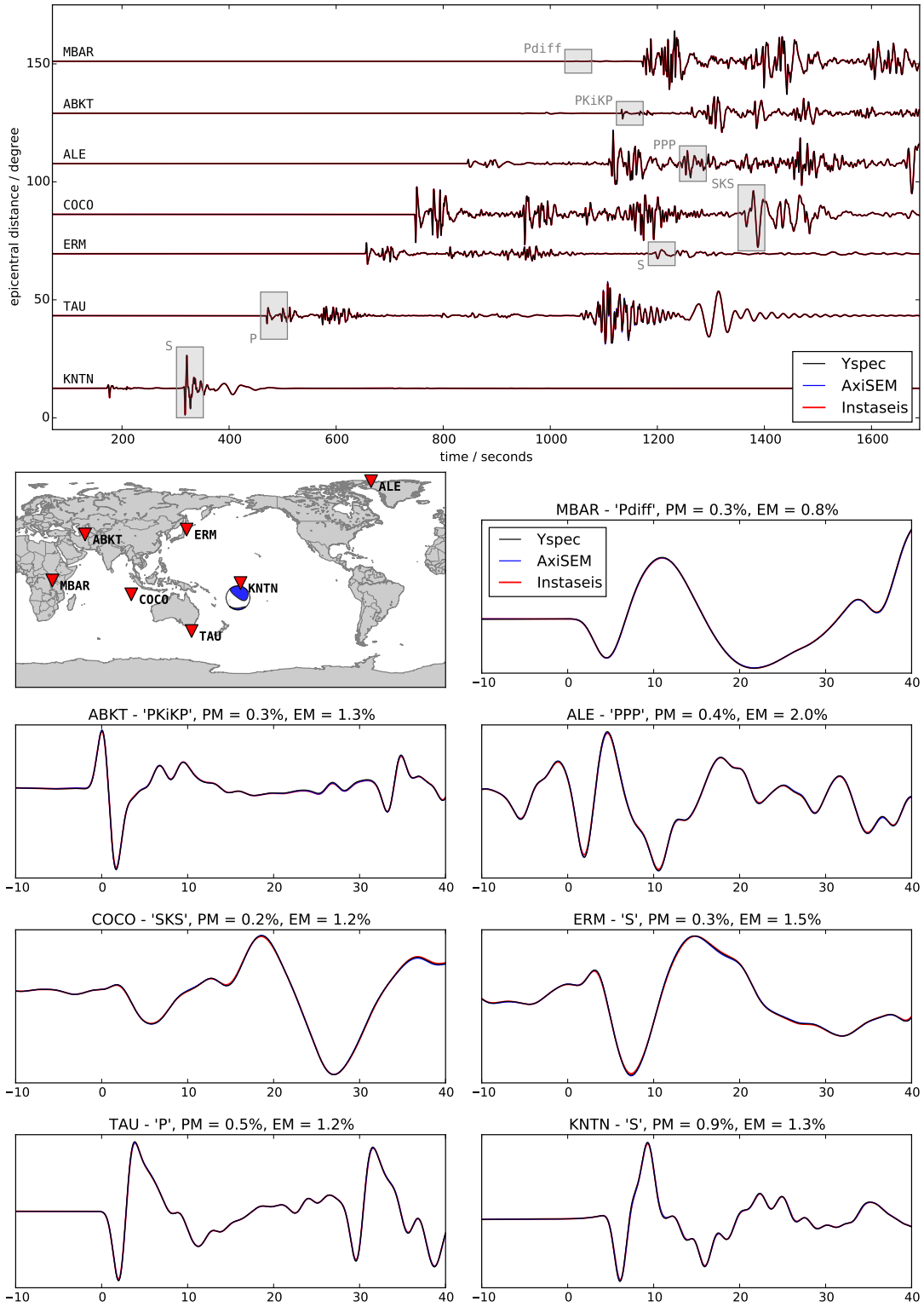

Figure A.12.: Comparison of vertical displacement seismograms (bandpass filtered from 50 s to 2 s period) for a moment magnitude Mw = 5.0 event in 126 km depth under the Tonga islands, computed with Instaseis, AxiSEM and Yspec in the anisotropic PREM model without ocean but including attenuation. The traces are recorded at the GSN stations indicated in the map. The zoom windows are depicted with gray rectangles in the record section and the time scale is relative to the ray-theoretical arrival. EM and PM (Kristekova et al., 2009) denote the envelope and phase misfit between Instaseis and Yspec traces in the corresponding time window.

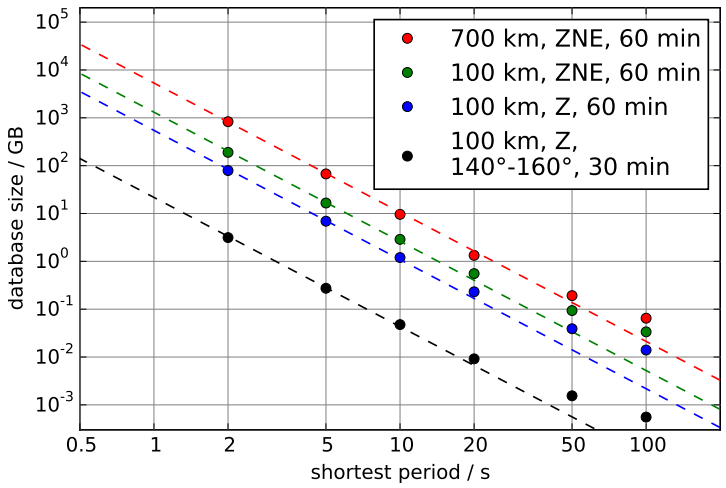

Figure A.13: Storage requirements of the reciprocal databases for PREM after zip compression for all three components and several parameter sets (maximum source depth, components, seismogram length and epicentral distance range). Dashed lines are fitted functions $g(f) = af^{2.7}$, where each point was weighted with the frequency f to ensure better fitting at the higher frequencies. The exponent is slightly smaller than the expected 3, because the zip compression is more efficient for longer time traces. At long periods, element sizes are governed by the layer thickness rather than the wavelength, resulting in the discrepancy from the power law at long periods.

A.4.2 Database Size

One major constraint for computing a database beside the CPU cost is the permanent storage requirement. Here, we summarize the most important parameters and the related scaling of the required disk space. The amount of data scales with the third power of the highest frequency resolved by the mesh, but zip compression is slightly more efficient for longer traces, resulting in an empirical exponent of 2.7, see Fig. A.13. Scaling with the length of the seismograms is slightly stronger than linear, again because the compression is more efficient on the zeros before the first P arrival. Scaling with depth and epicentral distance range is linear, where the prefactor for depth scaling is halved at each doubling layer of the mesh. The reciprocal databases for vertical (40%) and horizontal (60%) components are computed and therefore usable independently.

Several examples are shown in Fig. A.13: for Earth, a complete reciprocal database including all three components, all epicentral distances and sources down to 700 km and one hour of seismogram length accurate down to 2 s period, is about 1 TB in size. Calculating such a database once and storing it on a central server will give any user arbitrary and immediate access to short period synthetic seismograms without any further cost. More specialized databases are possible: for example to study inner core phases for shallow events in an epicentral distance from 140°to 160°, 200 GB storage suffices to store a database with a frequency of 2 Hz.

A.4.3 Performance

To evaluate the overall performance of Instaseis, two distinct parts have to be analyzed: First, the databases have to be generated with AxiSEM. Though very efficient, the database generation at short periods is a high performance computing task. However, AxiSEM scales well on up to 10,000 cores such that global wavefields can be computed at the highest frequencies within hours on a supercomputer. Detailed performance and scaling tests of AxiSEM can be found in Nissen-Meyer et al. (2014), here we just show the total CPU time required to compute full databases (i.e. horizontal and vertical component) for 1 hour long seismograms for two different time schemes (2nd order Newmark and 4th order symplectic Nissen-Meyer et al., 2008) and two planets (Earth and Mars) at a variety of resolutions, see Fig. A.14. The general

#CPUs	runtime	I/O time	Throughput	rel. I/O time
4624	1091 s	196 s	3.44 GB/s	18.0 %
2304	1802 s	281 s	2.40 GB/s	15.6 %
1152	2359 s	167 s	4.04 GB/s	7.0 %
576	4482 s	193 s	3.50 GB/s	4.3 %

Table A.1.: I/O performance for a typical setup of AxiSEM on SuperMUC. The simulation parameters were: 2 s shortest period, 3600 s simulation length, model: ak135f, vertical component, maximum source depth 700 km. The resulting uncompressed wavefield file has a size of 675 GB. The I/O throughput is not affected much by the number of CPUs involved. The throughput between different runs varies, which is probably caused by the changing I/O load on the system.

scaling of AxiSEM is proportional to T^{-3}, where T is the shortest period resolved by the mesh. The slight discrepancy from this power law at longer periods is due to the thin crustal layers causing a smaller global time step in the simulation. Simulations for Mars are approximately a factor five faster than for Earth, due to the smaller radius.

The performance of the second part, the seismogram extraction, on the other hand is rarely limited by raw computing power. It scales linearly with increasing frequency of the databases' Green's functions and can easily be accomplished on a standard laptop. The limiting factor in most cases is the latency of the storage system, e.g. the time until it starts reading from the database. To alleviate this issue we implement a buffering strategy on the functions reading data from the files: The Green's functions from a whole element of the numerical grid are read once and cached in memory. If data from the same element is needed again at a later stage it will already be in memory thus avoiding repeated disc access. Once the cache memory limit is reached, the data with the earliest last access time is deallocated, effectively resulting in a priority queue sorted by last access time. This optimization is very effective for most common use cases as they oftentimes require seismograms in a small range of epicentral distances and depths.

Instaseis comes with a number of integrated benchmarks to judge its performance for a certain database on a given system. The benchmarks emulate the computational requirements and data access patterns of some typical use cases like finite source simulations and source parameter inversions. Finite sources within the benchmarks are simulated by calculating waveforms for moment tensor sources on an imaginary fault plane along the equator ranging from the

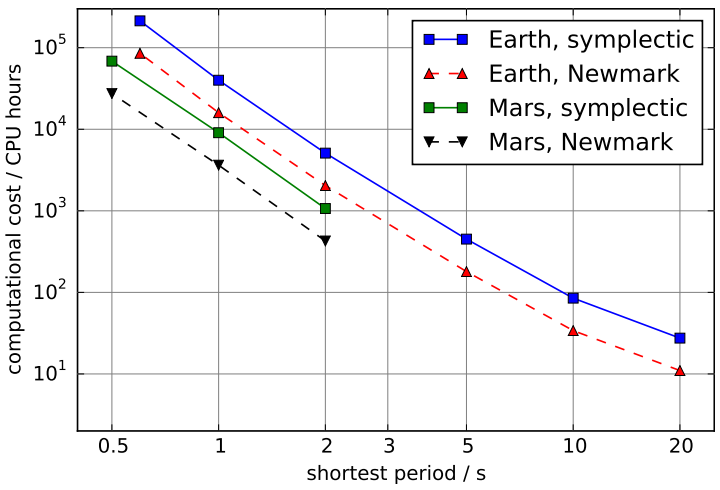

Figure A.14: Computational cost in CPU hours (measured on Monte Rosa, a Cray XE6, for Earth and Piz Daint, a Cray XC30, for Mars) to generate full Instaseis databases with 1 hour long seismograms for two time schemes: 2nd order Newmark and 4th order symplectic.

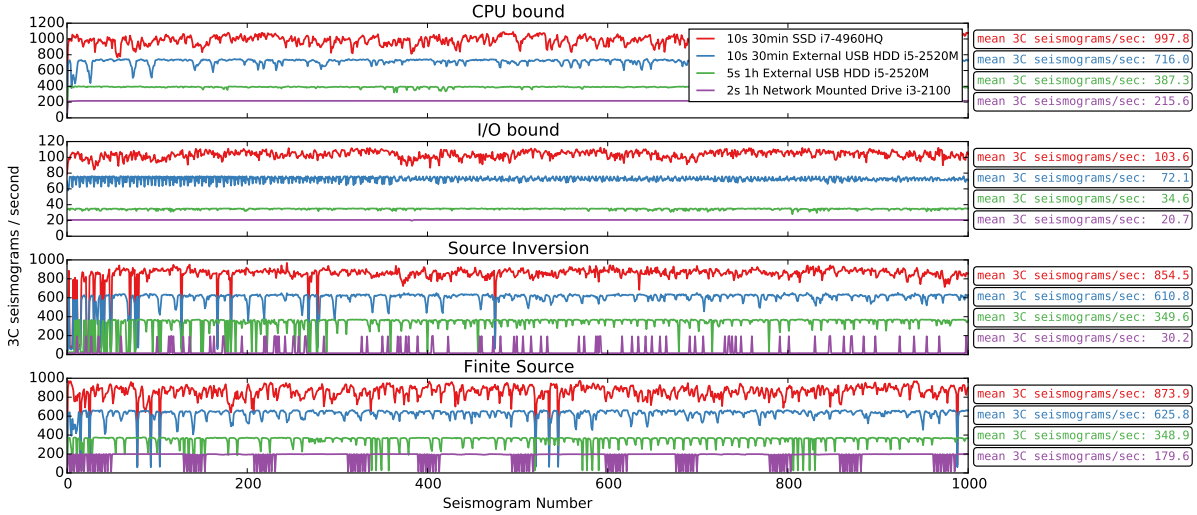

Figure A.15.: Results of benchmarks for four typical use cases run on different hardware with a variety of shortest periods. The graphs show the inverse time for the calculation of the *i*th three-component seismogram. Each run calculated 1000 three-component seismograms, is repeated ten times with the same random seed, the top and bottom values are discarded, and the mean of the remaining eight values is plotted. The CPU and I/O bound scenarios illustrate the speed with a fully efficient and a deactivated cache, respectively. The two bottom scenarios emulate real use cases, see the the main text for details. Amongst other things they show the consequence of a too small cache in the source inversion scenario for the 2 second run and the efficiency of the buffer in the finite source scenario for the same database.

surface to a depth of 25 km. One source is calculated for each kilometer in depth until the bottom of the fault is reached. This is repeated each kilometer along the fault's surface trajectory until the benchmark terminates. A source parameter inversion is simulated by calculating seismograms from moment tensor sources randomly scattered within 50 km distance to a fixed point. Results for four runs are shown in Figure A.15. As is the case with all benchmarks they have to be interpreted carefully, nonetheless they demonstrate the behaviour and performance characteristics of Instaseis on real machines.

A.5 Applications

In this section we depict several possible use cases of Instaseis. This list is not exhaustive and deliberately unconnected to provide a broad overview.

A.5.1 Graphical User Interface

To prominently highlight the features and nearly instantaneous seismogram extraction for arbitrary source and receiver combinations of Instaseis, we developed a cross-platform graphical user interface (GUI), shown in Figure A.16. It ships with the standard Instaseis package and is written in PyQt, a Python wrapper for the Qt toolkit.

Most evidently, this may be used for visual inspection and verification of any given AxiSEM Green's function database. Instaseis' performance permits an immediate visual feedback to changing parameters. This also delivers quantitative insight for an intuitive understanding of the features and parameter sensitivities of seismograms. Examples of this are the polarity flips of first arrivals when crossing a moment tensor's nodal planes, the triplication of phases for

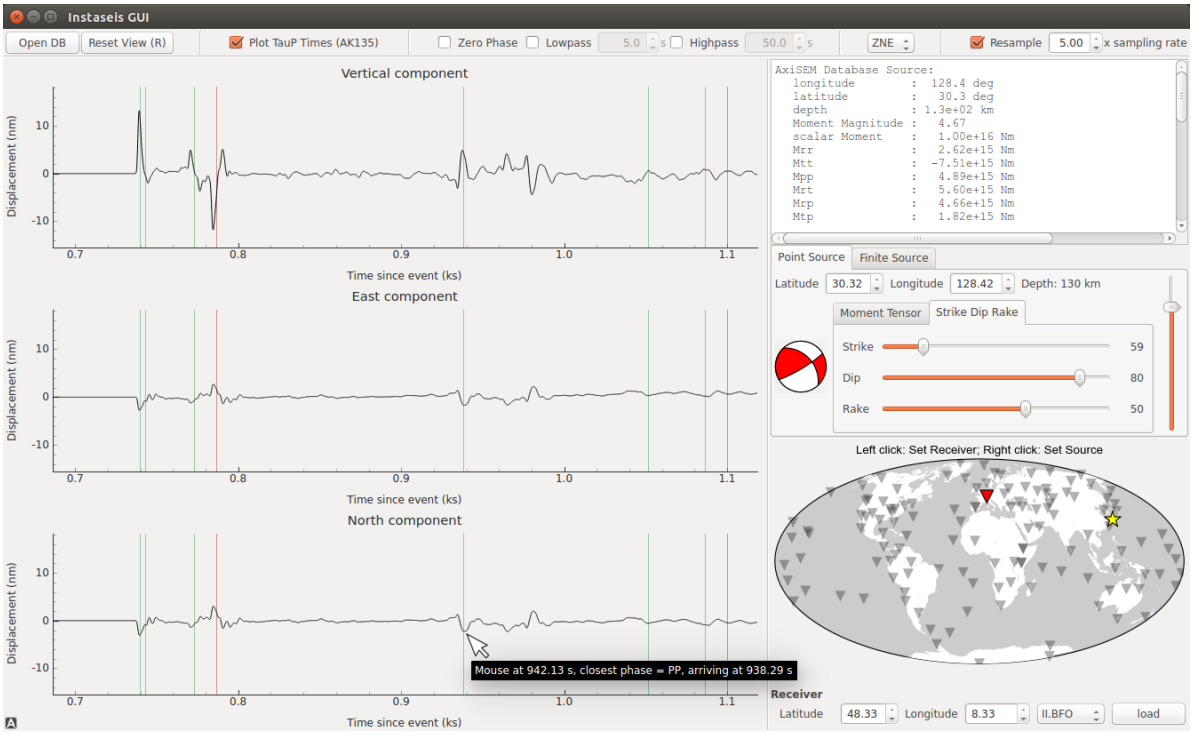

Figure A.16.: Screenshot of the Instaseis graphical user interface (GUI). Aside from quickly exploring the characteristics of a given Green's function database it is a great tool for understanding and teaching many features of seismograms. The speed of Instaseis enables an immediate visual response to changing source and receiver parameters. The left hand side shows three-component seismograms where theoretical arrival times of various seismic phases are overlaid as vertical lines. The bar at the top is used to change filter and resampling settings and the section on the right side is used to modify source and receiver parameters.

shallow sources, the Hilbert transformed shape of reflected phases and the relative amplitude of surface waves (especially overtones) depending on the earthquake depth. Furthermore, the GUI allows the calculation of seismograms from finite sources and the exploration of waveform differences in comparison to best-fitting points sources.

A.5.2 IRIS Web-interface

To enable usage of Instaseis seismograms to a broader community, we aim to remove all hurdles of computing and storing large databases locally. To this end, and in collaboration with IRIS, we plan to establish a webinterface to the Instaseis databases. In contrast to the ShakeMovie approach (Tromp et al., 2010), this interface will be able to handle arbitrary sources and receivers independent from catalogue data or other parameter limitations. The interface and databases will be described and benchmarked in detail in a separate publication, the status of this project can be viewed on `http://ds.iris.edu/ds/products/ondemandsynthetics`.

A.5.3 Finite-Frequency Tomography

In finite-frequency tomography (e.g. Nolet, 2008) information is extracted from recorded seismograms by matched filters in multiple frequency bands (Sigloch and Nolet, 2006; Colombi et al., 2014). A matched filter correlates a predicted signal with the measured signal to detect the predicted signal in the presence of noise. In the case of seismic tomography, a synthetic

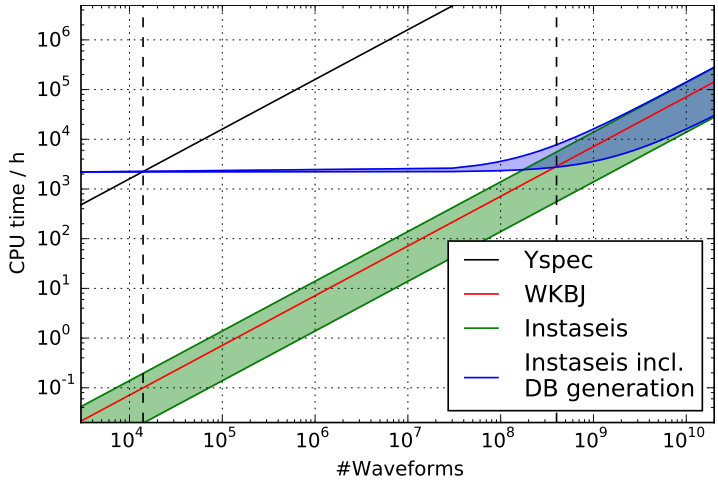

Figure A.17: Computational cost to compute many synthetic seismograms for finite-frequency tomography with a shortest period of 2 s using different methods. For Yspec we assume that for every source there are 1000 receivers with 3 components each. The shaded regions for Instaseis indicate the dependence of the performance on the actual source receiver distribution, compare Fig. A.15. Including the cost to generate the database with AxiSEM, Instaseis breaks even with Yspec for 14,000 waveforms, which is equivalent to about 5 sources in this configuration.

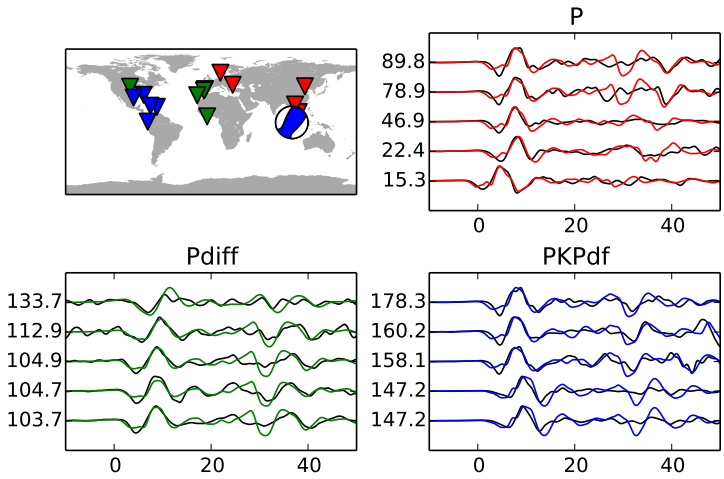

Figure A.18: Comparison between observed seismograms (black) and Instaseis synthetics for the Sumatra earthquake on September 30, 2009 with magnitude Mw 7.5 at 82 km depth. Vertical axis labels are epicentral distances, horizontal is time relative to the ray-theoretical arrivals. A Gabor filter with 3.7 s central period is applied to all traces and the synthetics are convolved with an inverted source time function (Sigloch and Nolet, 2006). The waveforms are aligned by computing relative time-shifts between data and synthetic seismograms using cross-correlation (similar to actual finite-frequency tomography).

seismogram is necessary, which is usually created by convolving a Green's function with an estimated source-time function. For body waves, short periods down to 1 s are commonly used (e.g. Stähler et al., 2012; Hosseini and Sigloch, 2015). Typical datasets contain thousands of earthquakes (e.g. Auer et al., 2014), each recorded at hundreds of stations, resulting in up to a million waveforms. For each of these waveforms, a separate Green's function has to be calculated, which requires solving the seismic forward problem at the desired frequencies. For wave propagation methods that solve the forward problem separately for each event, computing these reference synthetics presents a formidable computational challenge, which is why previous studies resorted to approximate solutions like WKBJ (Chapman, 1978) or the reflectivity method (Fuchs and Müller, 1971). The full method implemented in Yspec (Al-Attar and Woodhouse, 2008) is about an order of magnitude faster than AxiSEM in computing seismograms for a single source. However, at least in the current implementation the cost scales linearly with the number of events.

As Instaseis takes advantage of reciprocity of the Green's function, we can now build the whole database for all possible sources with only two runs of AxiSEM: one for the vertical and one for the horizontal components. Figure A.17 compares the computational cost of computing the reference synthetics down to 2 s period assuming that each event was recorded at 1000 three-component stations. Ignoring the cost of computing the database, Instaseis is comparable in performance to WKBJ, but actually returns full seismograms including all phases, see Figure A.18. In contrast to WKBJ, where each crustal reverberation has to be defined separately, it automatically calculates the full crustal response. Also, it appropriately models diffracted phases such as *Pdiff* and triplicated phases from upper mantle discontinuities. If we include the database generation, Instaseis breaks even in computational cost with Yspec already at about 14,000 waveforms, i.e. five events with 1000 three-component stations each. At about $5 \cdot 10^8$ waveforms, the cost of extracting the seismograms from the database becomes dominant over the database generation. Assuming 2000 seismograms per event, this is equivalent to 10,000 earthquakes, i.e. in the order of available earthquake catalogues. However, generating seismograms with different source locations or moment-tensor radiation patterns, which is often necessary in tomography, does not require a new database generation.

A.5.4 Probabilistic Source Inversion

Uncertainties in source parameters have been shown to have a strong influence on waveform tomography (Valentine and Woodhouse, 2010). Probabilistic point source inversion estimates the uncertainties of source parameters and their correlation. From these, the effect on seismic tomography can be estimated (Stähler and Sigloch, 2014). It requires the repeated calculation of synthetic waveforms for varying moment tensors, depths and source time functions to calculate the likelihood and posterior probability density of models in a Bayesian sense. Changing source time function and moment tensor is extremely efficient from an Instaseis perspective, and the limitation to a fixed epicenter means that the I/O buffering can be done very efficiently, which is reflected in the *Source Inversion* testcase in the benchmark (Fig. A.12).
From a previous study (Stähler and Sigloch, 2014), we assume that for an inversion for depth, the moment tensor and the source time function, a 20-dimensional model space has to be sampled, which requires to perform roughly 60,000 forward simulations. Using 100 seismic stations and three-component seismograms, this means that roughly $1.8 \cdot 10^7$ waveforms have to be calculated for one source inversion, costing on the order of 50-100 CPU hours (Fig. A.17).

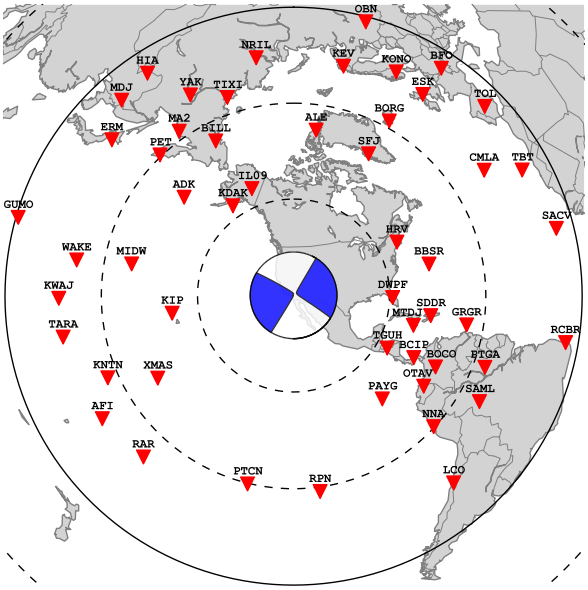

Figure A.19.: Stations used in the Source Inversion Validation (SIV) exercise. Circles mark 30°, 60° and 90° epicentral distance. The finite source is a M 7.8 strike-slip earthquake in southern California represented by $\approx 10^5$ point sources, the beachball represents the centroid moment tensor, i.e. the orientation and predominant direction of slip of the overall fault.

A.5.5 Finite Sources

Finite sources can be represented in Instaseis by a cloud of point sources without limitations on the fault geometry or source time functions. Each point source needs to be attached with a moment tensor, a sliprate function and a time shift relative to the origin time. These can for instance be retrieved from a standard rupture file (.srf). As a show case, we computed the seismograms for the source inversion validation (SIV) exercise #3 (`http://equake-rc.info`). The source is a M 7.8 strike-slip earthquake on the San Andreas Fault represented by $\approx 10^5$ point sources, where each source has a different mechanism and sliprate function. The 52 stations are in 30°to 90°epicentral distance (see Fig. A.19), where the P wave arrival is supposed to be well separated (compare Fig. A.1). Excluding the cost of generating the database, it cost a total of 12 CPU hours to compute the 52 one hour long three-component seismograms accurate down to 5 s.

Fig. A.20 compares the Instaseis seimograms to P-phases computed with the frequency-wavenumber integration method (fk) by Kikuchi and Kanamori (1982), where only direct and surface reflected phases where taken into account. While the first arriving waves agree to certain extent with Instaseis providing systematically larger amplitudes, there are significant differences for later time windows. These are due to additional phase arrivals within the time window (especially triplicated PP, compare Fig. A.1) and crustal reverberations not modeled by the fk method. For events with long rupture durations as in this example (200 s) this suggests that more accurate waveforms should be beneficial for finite source inversions.

A.5.6 Insight / Mars

The upcoming NASA-lead Mars Insight mission (Banerdt, 2013), to be launched in March 2016 and scheduled to land September 2016, will deploy a single station with both a broad-

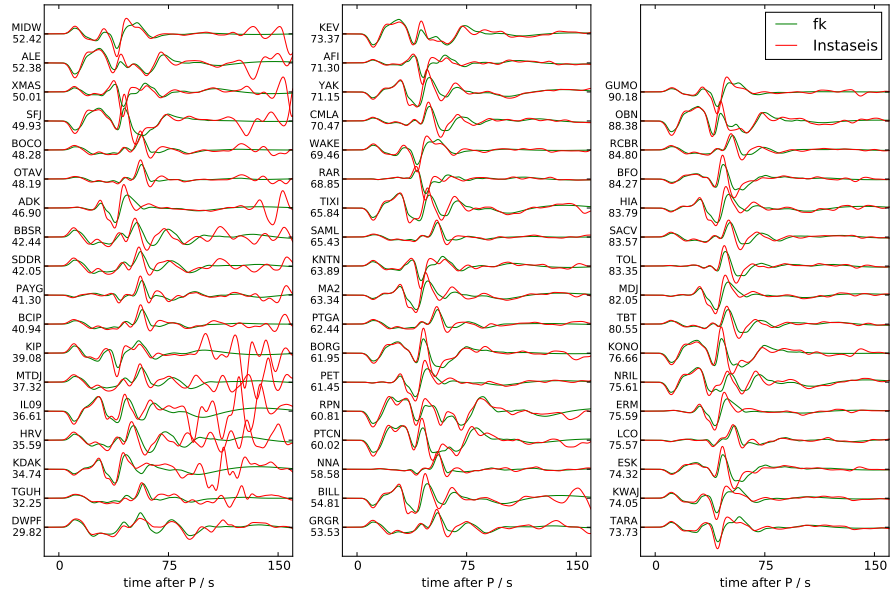

Figure A.20.: Seismograms for the SIV benchmark, Z-component aligned on the P arrival band-pass filtered between 5 and 100 s period. The labels denote the station code and epicentral distance. In the frequency-wavenumber integration (fk, Kikuchi and Kanamori, 1982), only direct P and the depth phases were included, while Instaseis provides full seismograms, including PP, PcP and other phases. Especially for the stations in less than 40°distance, the effect is profound, since PP arrives as a triplicated complex wave train only 70-100 s after P. Due to the long source duration, the PP arrival overlaps with the direct P wave train for several stations.

band and a short period seismometer on Mars. This will be the first extra-terrestrial seismic mission since the Apollo lunar landings (1969–1972) and Mars Viking missions (1975) with the goal of elucidating the interior structure of a planet other than Earth. The instrument will record local, regional, and more distant marsquakes, including meteorite impacts, and send data back to Earth for analysis.

Our knowledge of the seismic structure of Mars is limited, because of lack of resolution of currently available areophysical data (e.g., Khan and Connolly, 2008) and the limited sensitivity of the Viking seismometers due to their installation on board of the lander. For this reason, we will generate databases of "reference" seismic waveforms for a comprehensive collection (order of magnitude 1000) of 1-D Martian models to be used by modelers and analysts in preparation for the Insight mission. The models are constructed from current areophysical data (mean mass, mean moment of inertia, tidal Love number, and tidal dissipation) and thermodynamic modeling methods and summarize our current understanding of the internal constitution of the planet. AxiSEM and hence Instaseis can readily be used to propagate waves on Mars, see Fig. A.21, allowing us to build these databases very efficiently.

A.5.7 Synthetic Ambient Seismic Noise

As mentioned in section A.2.3, seismograms generated by force sources can be extracted from the same reciprocal databases. This is particularly interesting for studying ambient seismic noise. By cross-correlating noise recorded at two stations, using long enough time series and under certain assumptions (uncorrelated, isotropically distributed white noise sources), it is possible to retrieve the Green's function of the medium between the two stations (e.g. Sanchez-Sesma, 2006; Gouédard et al., 2008). However, not all of these assumptions are met in nature,

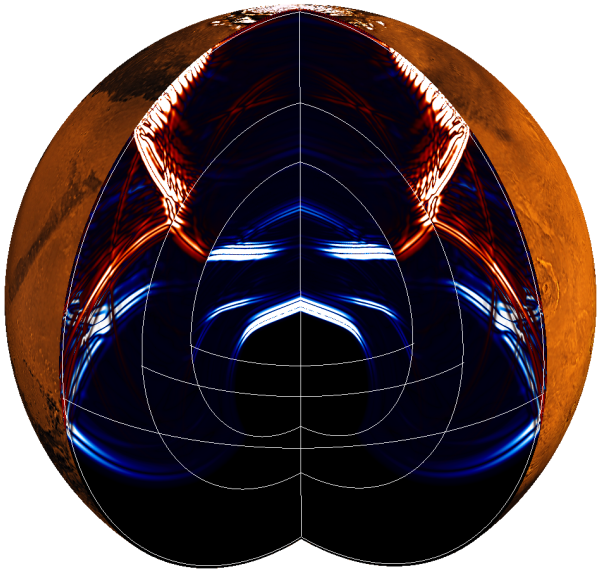

Figure A.21.: Seismic waves travelling in Mars after a meteorite impact at its north pole computed with AxiSEM. P-waves are shown in blue and S-waves and surface waves in red.

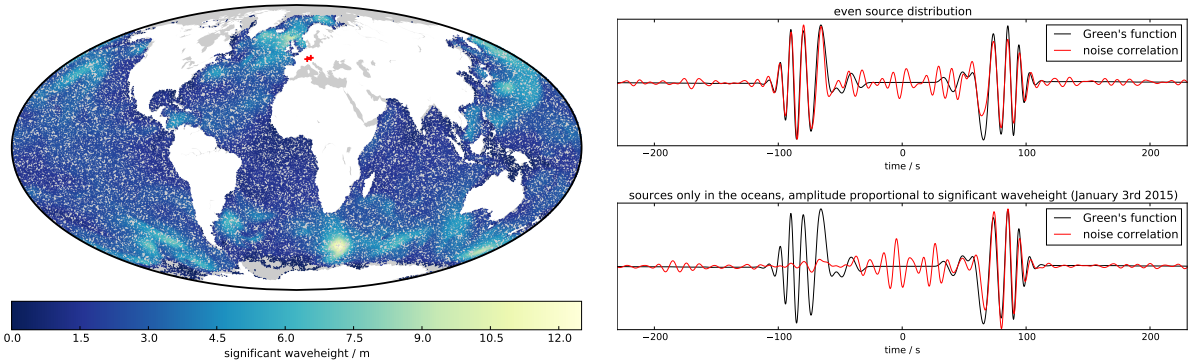

Figure A.22.: Ambient seismic noise cross correlations computed with Instaseis. Left: 100,000 vertical force sources located in the oceans and amplitude proportional to the significant wave height from the NOAA WAVEWATCH III model on Jan. 3rd, 2015 (Tolman, 2009). Red crosses indicate the receivers located in Munich and Zurich. Right: Cross correlations of 20 days of noise for (top) evenly distributed noise sources and (bottom) the sources in the map, the traces are normalized to their maximum amplitude.

e.g. the noise sources are not evenly distributed (Tsai, 2009; Froment et al., 2010; Basini et al., 2013). Also, the noise sources themselves are not yet well understood, especially with respect to the generation of Love waves in the microseismic band (Nishida et al., 2008).

Instaseis provides a basis to quickly generate noise synthetics to study such effects, which we illustrate in Fig. A.22. We computed noise cross correlations, accurate down to a period of 5 s, for a total of 20 days of noise data generated with 100,000 noise sources. The calculation only took 1 CPU hour. In the first case, the noise sources consist of vertical forces with a random source-time function, all have the same amplitude and are distributed evenly on the globe. The resulting cross correlation is in good agreement with the Green's function, which is obtained by introducing an impulse source at each of the stations in Zurich and Munich. In the second case, sources are located in the oceans only, their amplitude proportional to the significant wave height (Gualtieri et al., 2013). For the two stations located in Zurich and Munich the close sources are thus solely located in the west, which leads to strong asymmetry in the retrieved

correlations (Stehly et al., 2006). Instaseis thus enables users to study noise on the global scale across the microseismic band, by generating realistic waveforms at negligible cost.

A.6 Conclusion

In this chapter we presented a readily available methodology and code to extract seismograms for spherical earth models from a Green's function database. High efficiency in the generation of databases and very fast extraction (on the order of milliseconds per seismogram) of highly accurate seismograms (indistinguishable from conventional forward solvers) can then replace previously employed approximations such as WKBJ, reflectivity or frequency-wavenumber integration methods that were used for computational reasons in many applications of global seismology. Instaseis is open source and available with extensive documentation at `www.instaseis.net`.

Future developments include Cartesian local domains with layered models, which are not yet supported by AxiSEM. As a large fraction of earthquakes are located below oceans and receivers on continents, it may be beneficial for body waves studies to take advantage of the axisymmetric capability of AxiSEM and place the receiver on a circular 'island' of continental crust within a global oceanic crustal model.

Author contributions

M. v. D. and L. K. implemented Instaseis, M. v. D., S. C. S. and T. N.-M. continuously develop AxiSEM and added the database output, and K. H. prepared the finite-frequency example. M. v. D. prepared the manuscript with contributions from all co-authors.

Acknowledgements

We thank Alex Hutko and Chad Trabant (IRIS) for valuable discussions on the database selection and implementation as well as user demand. Amir Khan provided the Mars model and Olaf Zielke the reference data used in Fig. A.20. Celine Hadziioannou supported us to design the noise example. We gratefully acknowledge support from the European Commission (Marie Curie Actions, ITN QUEST, `www.quest-itn.org`) and the EU-FP7 VERCE project (number 283543). Data processing and downloading was done using ObsPy (Megies et al., 2011; Beyreuther et al., 2010) and ObsPyLoad (Scheingraber et al., 2013). Computations where performed at the ETH central HPC cluster (Brutus), the Swiss National Supercomputing Center (CSCS), the UK National Supercomputing Service (HECToR) and the Leibniz Supercomputing Center (LRZ), whose support is gratefully acknowledged.

B
Syngine

Syngine (`http://ds.iris.edu/ds/products/syngine/`, last accessed March 2017), is an extension of the previously introduced Instaseis software that runs as a webservice at the IRIS DMS.

It offers on-demand and custom tailored seismograms served over HTTP. The free service produces full seismic waveforms including effects like attenuation and anisotropy that are calculated in commonly used spherically symmetric Earth models (PREM, ak135-f, iasp91). Users can freely adjust sources and receivers, retrieve seismograms from finite sources, convolve with arbitrary source time functions, and download Green's functions suitable for moment tensor inversions. Syngine extracts and processes seismograms in as fast as fractions of a second making it suitable for applications demanding short iteration times and a large number of waveforms.

For the first time, researchers without large computational resources or specialized knowledge can easily access high-quality, custom, broadband seismograms. In this chapter we present the rationale and basic principles of our method, including its limitations. Additionally we demonstrate the features of Syngine and the included Earth models, showcase several applications, and discuss future possibilities. A version of this chapter is published as:

Krischer, L., Hutko, A., van Driel, M., Stähler, S., Bahavar, M., Trabant, C., & Nissen-Meyer, T. (2017).
On-demand custom broadband synthetic seismograms
Seismological Research Letters, 88(4)
`https://doi.org/10.1785/0220160210`

B.1 Introduction

Synthetic seismograms are fundamental for a plethora of tasks in seismology, most notably for comparing to observed seismograms and thus testing hypotheses. A wide array of methods to calculate these synthetics have thus developed over the decades. They range from being very accurate but expensive to calculate to being very fast but using heavy approximations that might not be acceptable for a given purpose. Additionally, many seismological software packages are difficult to use, requiring significant expert knowledge to acquire trustworthy results, especially when significant changes to input parameters are required, such as new Earth models.

In this chapter, we introduce Syngine (`http://ds.iris.edu/ds/products/syngine/`), a web service to calculate custom-tailored, accurate seismograms sent to users upon request with return times as fast as fractions of a second. Syngine delivers full seismograms for spherically symmetric Earth models with anisotropic and viscoelastic rheologies, but the current set of databases does not include the effect of self-gravitation and Earth rotation, which might affect long period applications. The source-receiver geometry can be arbitrarily chosen under the constraint that the receiver is at the surface of the Earth. Further features include seismograms from finite sources, convolutions with arbitrary source time functions, the download of independent components to construct arbitrary source mechanisms with simple calculations, and windowing traces around seismic phases. A major focus of the project is usability, reliability, trustworthiness, and accuracy of the calculated seismograms.

The sole prerequisite to using Syngine is internet access. Thereby, users will not have to perform an expensive numerical simulation or store very large databases. This equips anyone, including researchers and groups that had no previous access to synthetic waveforms, either due to lack of computational facilities or knowledge, with access to broadband, high-accuracy, full synthetic seismic waveforms.

This chapter is organized as follows: In section B.2 we introduce the mathematical, numerical, and technical methodology used to generate the synthetic seismograms. Section B.3 presents the features of the Syngine service and section B.4 discusses the rationale behind the offered Earth models providing guidance for their use. Finally we discuss potential applications in section B.5 and the big picture in section B.6. The code to (re)create all figures except figures B.7 and B.9 in the form of interactive Jupyter notebooks (Pérez and Granger, 2007) can be found on `http://seismo-live.org`.

B.2 Methodology

In this section we present a short and intuitive introduction to the numerical and mathematical methods used to generate and extract the synthetic seismograms. The fundamental methodology has been described at length in the literature and we will refer interested parties to relevant works where appropriate. We aim to provide readers the necessary knowledge to understand the capabilities and also the limitations and trade-offs of our scheme, enabling them to judge its applicability to arbitrary problems. In a nutshell the response of the medium to two forces - a vertical and a horizontal force source at the surface of the Earth - is simulated and recorded in a two dimensional domain. By exploiting the reciprocity of the Green's functions it is possible to swap source and receiver and, assuming a spherically symmetric medium, reconstruct that response between any two points on the planet given that the receiver is positioned at a fixed

radius. In most instances this will either be the surface of the Earth or the bottom of the oceans. The method employed by Instaseis (`http://instaseis.net`, van Driel et al. (2015)) assures that the seismic wavefield is a full solution to the elastic wave equation in three dimensions, the only limitation is the required spherical symmetry of both the elastic properties and the domain.

B.2.1 Generation of the Waveform Databases

The first step is to compute databases of accurate waveforms using AxiSEM. AxiSEM (`http://axisem.info`, Nissen-Meyer et al. (2007b, 2014)) employs a spectral element scheme (e.g. Komatitsch and Tromp, 1999) to propagate global seismic waves in axially symmetric media. This assumption allows the analytic decomposition of the 3-D wavefield into several 2-D wavefields. Thus only a 2-D numerical problem must be solved which is orders of magnitudes cheaper and therefore enables the calculation and storage of global wavefields at high frequencies. The final 3-D wavefield nonetheless contains all effects of a viscoelastic rheology like attenuation (van Driel and Nissen-Meyer, 2014a) and anisotropy (van Driel and Nissen-Meyer, 2014b). An important limitation to keep in mind is that the seismograms only capture the relevant physics up to a period of about 100 seconds. At longer periods, effects such as gravitation and Earth rotation play an increasingly important role (see e.g. Komatitsch and Tromp (2002b)) but are, at this point, not taken into account in AxiSEM. The overall effect of the neglected physics can be comparatively small but in some scenarios it does matter and it is thus crucial to be aware of it. The short-period bound is dictated by the numerical mesh and can vary for each database. Strikingly, for a full database encapsulating all possible source-receiver geometries, only two simulations are required due to reciprocity in the wave equation (van Driel et al. (2015)): one for a vertical force source and one for a horizontal force source, both located at the surface of the planet. The displacement response for each is stored at every point of the numerical simulation grid up to a maximum depth. There is no limitation to the recorded depth range but greater ranges result in significantly increased database sizes. As the recorded range determines the possible source depths it is natural to limit it to the maximum depth of naturally occurring seismicity.

B.2.2 Seismogram Extraction

Exploiting the reciprocity now allows one to place a source anywhere in the recorded region where the initial force sources correspond to the different components of a receiver. Recalling that we simulated in a spherically symmetric medium, we can place the reciprocal receiver anywhere on the planet and extract its response to sources throughout the recorded depth range. The stored displacements allow the on-the-fly reconstruction of the full strain tensor and resulting seismograms therefore have all three components of a fully 3-D wavefield. Storing the displacement on the grid of the numerical simulation and subsequent on-the-fly evaluation of the same basis functions as used in AxiSEM means that the spatial interpolation adds no additional error and is as accurate as the spectral-element simulation for arbitrary source-receiver locations independent of the numerical grid. The wavefields are temporally downsampled to four samples per mesh period which captures the spectral content of the wavefield down to around -80dB with respect to velocity. Seismograms are later upsampled again with a tapered sinc function which approximates an optimal reconstruction filter. This extraction is performed by

Instaseis, the final seismograms are amended with meta information and serialized to common file formats by ObsPy (`http://obspy.org`, Megies et al. (2011); Krischer et al. (2015b)).

This combination of tools allows the extraction of broadband three-dimensional seismograms propagating through spherically symmetric spheres in as little as fractions of a second with negligible spatial and temporal interpolation errors. Downsides of Instaseis as a standalone tool are the demanding computational cost for each user to generate the initial AxiSEM databases, and their unwieldy file sizes. Syngine, the topic of this chapter, tackles both these problems.

B.3 Features

Syngine is a web service offering convenient and fast access to synthetic seismograms, as described in the previous section, over the HTTP protocol. Waveforms can be downloaded one at a time or in bulk.

B.3.1 Seismograms

Seismograms constitute the very core of Syngine. Each seismograms represents the response of the medium at one receiver to a particular source. Syngine is used by constructing a web URL that encodes the source, the receiver, and additional optional parameters. Accessing this URL either with a script or a web browser triggers the server-side extraction of the requested seismogram(s) which are then sent to the users. All available parameters are presented in detail in table B.1, a few of them are described in more detail in this section. Parameters are roughly grouped into source, receiver, and miscellaneous parameters.

Sources always need a location specified by geographic latitude, longitude, depth values and a mechanism. Mechanisms can be defined by passing the 6 independent moment tensor components M_{rr}, M_{tt}, M_{pp}, M_{rt}, M_{rp}, and M_{tp} or by giving strike, dip, rake, and the scalar seismic moment M_0. Alternatively the source mechanism can be a vectorial single force which is useful, for example, for noise studies (see e.g. Basini et al., 2013)). For convenience all of these parameters can be replaced by passing an event identifier from an event catalog. Syngine at the time of writing supports event source lookups in the GCMT catalog (Ekström et al., 2012) and the USGS finite fault model database; other catalogs may be added in the future.

Receivers also require coordinates and must always be located at a fixed radius depending on the database, usually the surface of the Earth. Thus only latitude and longitude coordinates are required for a request. Network and station codes can be passed instead and they will be used to locate coordinates of the corresponding real stations in the IRIS database. Wildcards can be applied to download a large number of synthetics at once. The following query for example will download seismograms for all stations starting with `A` from the virtual GSN network: `...&network=_GSN&station=A*&...`

Convenience parameters such as a data scaling factor or a custom file name label are grouped under miscellaneous. The Earth model from which to extract seismograms is also given as a parameter. It specifies the velocity model as well as the frequency content of the database. Additionally any combinations of vertical, north, east, radial, and transverse receiver components can be requested either in displacement, velocity, or acceleration, each in SI units. Last but not least the seismograms can be resampled to any sampling rate larger than the database sam-

Model parameters		
`model`	`prem_a_5s`	Velocity model.
Output parameters		
`format`	`miniseed`	Output file format
`label`	`Tohoku`	Label to be included in file names.
`components`	`ZRT`	Seismogram components. Any of `Z`, `N`, `E`, `R`, and `T`.
`units`	`velocity`	Units: `displacement`, `velocity`, or `acceleration`.
`scale`	`3.3`	Amplitude scaling factor.
`dt`	`0.2`	Sampling interval in seconds.
`kernelwidth`	`8`	Width of the resampling kernel.
`sourcewidth`	`15`	Optionally convolve with a Gaussian STF given the width in seconds.
Time parameters		
`origintime`	`2010-02-27T06:34:14`	Source origin time as absolute date and time.
`starttime`	`2010-02-27T06:34:14`	Start time of the synthetic traces.
`endtime`	`2010-02-27T06:34:14`	End time of the synthetic traces.
Receiver parameters		
`network`	`IU`	Network code to lookup station coordinates.
`station`	`ANMO`	Station code to lookup station coordinates.
`receiverlatitude`	`47.6`	Receiver latitude in degrees on WGS84.
`receiverlongitude`	`-122.3`	Receiver longitude in degrees on WGS84.
`networkcode`	`YY`	Directly set network code.
`stationcode`	`D1Z1`	Directly set station code.
`locationcode`	`A1`	Directly set location code.
Source parameters		
`eventid`	`GCMT:C201002270634A`	Event identifier to lookup earthquakes.
`sourcelatitude`	`-89.99`	Source latitude in degrees on WGS84.
`sourcelongitude`	`-179.99`	Source longitude in degrees on WGS84.
`sourcedepthinmeters`	`699000`	Source depth in meters.
Source mechanism as a moment tensor		
`sourcemomenttensor`	`1e22,-3e22,-1e22, 3e22,-1e22,9e21`	Moment tensor source as M_{rr}, M_{tt}, M_{pp}, M_{rt}, M_{rp}, M_{tp} in Nm.
Source mechanism as a double couple		
`sourcedoublecouple`	`19,18,116,1e19`	Double couple source as strike, dip, rake, and the scalar seimic moment in Nm.
Source mechanism as forces		
`sourceforce`	`1e22,1e22,1e22`	Vectorial force source as F_r, F_t, and F_p in N.

Table B.1.: Available parameters for the seismograms service of Syngine. Please see the website for a detailed and up-to-date explanation of them. A couple are also elaborated upon in the text of this chapter and a full example to request vertical component data for the IU.ANMO station and the 2002 Denali earthquake is: `http://service.iris.edu/irisws/syngine/1/query?network=IU&station=ANMO&components=Z&eventid=GCMT:M110302J`

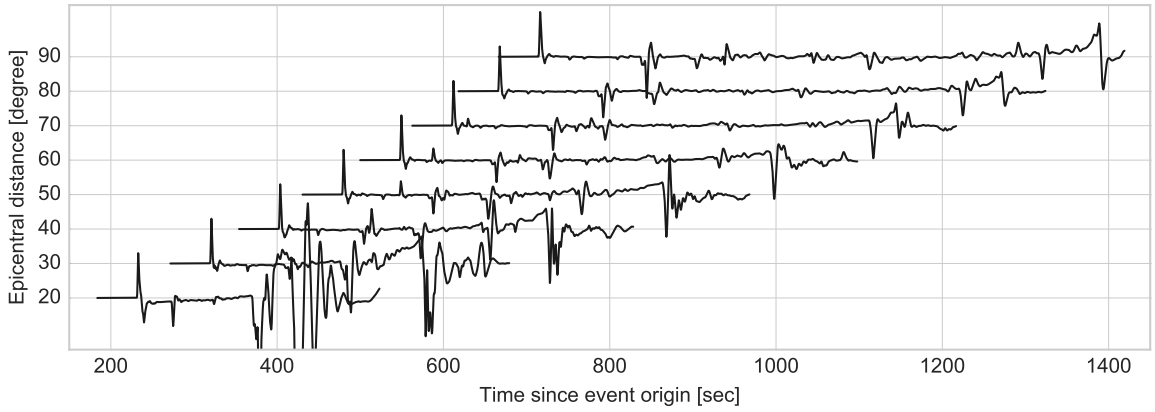

Figure B.1.: Record section demonstrating a request with phase relative start and end times. This figure shows a plot of data from 8 stations at various epicentral distances. Each vertical component trace contains data from 50 seconds before the theoretical P wave arrival to 100 seconds after the theoretical S wave arrival. The source is a pure M_{rr} source at 600 km depth, the earth model is ak135-f (Kennett et al., 1995; Montagner and Kennett, 1996), and the database contains periods from 2 to 100 seconds. Each trace is normalized around the first arrival to account for attenuation and geometrical spreading. The phase relative time settings can greatly reduce the amount of data that has to be transferred. Generating this plot, including the database extraction, serialization, sending the data to another continent, and actually plotting it takes only a few seconds.

pling rate. The resampling employs a very high quality reconstruction algorithm and is careful to avoid edge effects, thus we encourage users to make use of this functionality.

Seismograms are returned either as a ZIP archive of SAC (Helffrich et al., 2013) files or as a single MiniSEED (Incorporated Research Institutions for Seismology (IRIS), 2012) file. Both have all their important header values filled to aid in data organization and mimicking recorded data.

B.3.2 Geographical Coordinates

Exploiting reciprocity necessitates the use of a spherically symmetric planet unlike the real Earth and Instaseis as well as AxiSEM internally use a spherical geocentric coordinate system. Most seismological applications and data sets on the other hand define geographical coordinates on the WGS84 ellipsoid (National Imagery and Mapping Agency, 2000) and we follow that convention and adapt it to our internally used spherical symmetry.

All geographical coordinates that users specify are assumed to be given in WGS84 and are, prior to seismogram extraction, converted to spherical, geocentric coordinates. This effectively is a zeroth order ellipticity correction for comparison with real Earth data. Coordinates passed back to the users, for example in the SAC file headers, are also in WGS84. This only affects the latitude which is usually defined as the angle between the surface normal at a point and the surface normal at the equator at the same meridian. This differs for spherical and elliptical coordinate systems. The difference is usually fairly small and many applications are not affected by it, but it is vital to understand how Syngine handles coordinates.

Custom source time function parameters		
`cstf-relative-origin-time-in-sec`	`5.0`	Offset in seconds to set the origin time of the of the seismogram.
`cstf-sample-spacing-in-sec`	`10.0`	Sample interval of the STF in seconds.
`cstf-data`	`0.0,0.5,0.0`	The custom STF data values.
Source parameters		
`ffmeventid`	`USGS:us20002926`	Event identifier to lookup finite source model.

Table B.2.: Additional parameters for the seismogram service which allow the specification of a custom source time function and for finite source calculations. All other, non source related parameters are identical to the seismograms service and are listed in table B.1.

B.3.3 Phase Relative Times

A multitude of applications only require information about certain seismic phases. To this end, the Syngine service supports the specification of start and end times relative to any phase arrival. We use a port of the TauP Toolkit (Crotwell et al., 1999) in ObsPy (Beyreuther et al., 2010) which employs the $\tau - p$ method (Buland and Chapman, 1983) used to calculate theoretical arrival times. It supports arbitrary phase names within its syntactical and semantic limitations and performs the calculations in a spherical coordinate system for better accuracy compared to earlier implementations. Generating phase-windowed seismograms can greatly reduce the amount of data that needs to be transferred. Figure B.1 shows a record section with phase relative time settings.

B.3.4 Source Time Functions

Syngine relies on synthetics calculated with the spectral element method. Sources for such simulations cannot be true delta functions as this is numerically unstable. A narrow Gaussian smoothly resembling a delta peak is used instead. Considering the band-limited nature of the signal, this is a good approximation. Also, each database contains contains a time series of the slip and slip rates used in AxiSEM to create it. Calculating seismograms with new slip rate/source time functions (STF) requires the deconvolution of the database's STF from the signal followed by a convolution with the desired STF. Performing this in a single step yields numerical stability under some assumptions regarding the frequency content of both STFs: The new STF must not contain frequencies that the database STF does not have. We disallow STFs that have a finer sampling than the database STF which contains all simulated frequencies and thus almost guarantees this. If the need arises we'll implement some further stabilizing algorithms.

New seismograms u_j having an STF s_j are calculated from the database seismograms u_i with the database STF s_i as follows:

$$u_j = \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(w\ u_i\right)\mathcal{F}\left(s_j\right)}{\mathcal{F}\left(s_i\right)}\right) \tag{B.1}$$

$\mathcal{F}$ and $\mathcal{F}^{-1}$ are the forward and inverse (fast) Fourier transforms, and w is a tapering function to make sure the initial seismogram ends with zero.

Syngine can perform this process, which we refer to as reconvolution, in a stable manner that also deals with subtle issues such as time shifts. Users can either upload their own source time function (please see table B.2 and the Syngine documentation for instructions) or use the `sourcewidth` parameter which reconvolves the seismograms with a Gaussian defined as

$$s = \frac{4}{a\sqrt{\pi}} e^{-\frac{16}{a^2}(t-t_0)^2} \tag{B.2}$$

with t being the time, t_0 the offset, and a the source width. See figure B.2 for plots of this function as well as its effect on seismograms. The peak of the Gaussian will be at the origin time, this results in no apparent time shift but some acausal effects. This for example can be used to approximate the triangular moment rate functions used in the GCMT catalog (Ekström et al., 2012) in which case the `sourcewidth` parameter is twice the half-duration specified in the catalog.

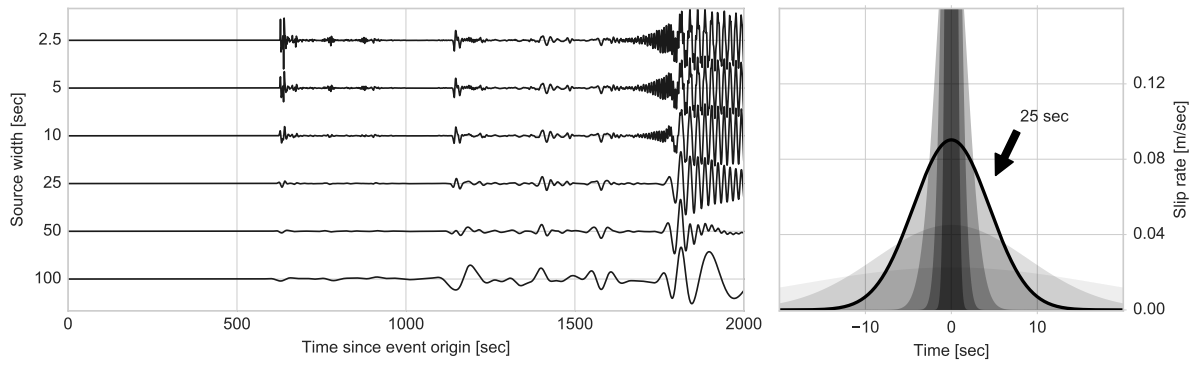

Figure B.2.: Illustration of the effect of the `sourcewidth` parameter. The left panel shows seismograms convolved with Gaussian source time functions of different source widths. The seismograms are normalized as the source time function has a profound impact on the amplitude. Note the acausal effects - the event origin time is always defined to be the peak of the Gaussian. The right hand side plots the used source time functions in six shades of gray. The black outline shows the Gaussian with a source width of 25 seconds.

B.3.5 Finite Source Seismograms

Although very convenient it is oftentimes insufficient to describe seismic wavefields as originating from single points in space. Real earthquakes, especially larger ones, have rupture surfaces and the point source approximation breaks down. The superposition principle applied to linear elastodynamics allows the calculation of finite-source seismograms for kinematic sources via summation of a number of distributed point sources, each with their independent rise time. Finite source descriptions usually originate from kinematic source inversion, but finite-source models coming from dynamic rupture simulations are emerging as well. The SRCMOD database (Mai and Thingbaijam, 2014) collects Finite Fault Models (FFM), other groups such as the US Geological Survey (USGS) offer FFM solutions for download. Syngine supports the calculation of seismograms from FFMs. As of the writing of this chapter it can use models described in the USGS "param" file format; other formats might be supported in the future.

The parameters are largely similar to the standard Syngine functionality for seismograms. Phase-relative times are calculated from the sources' hypo-centers, e.g. the patch with the first

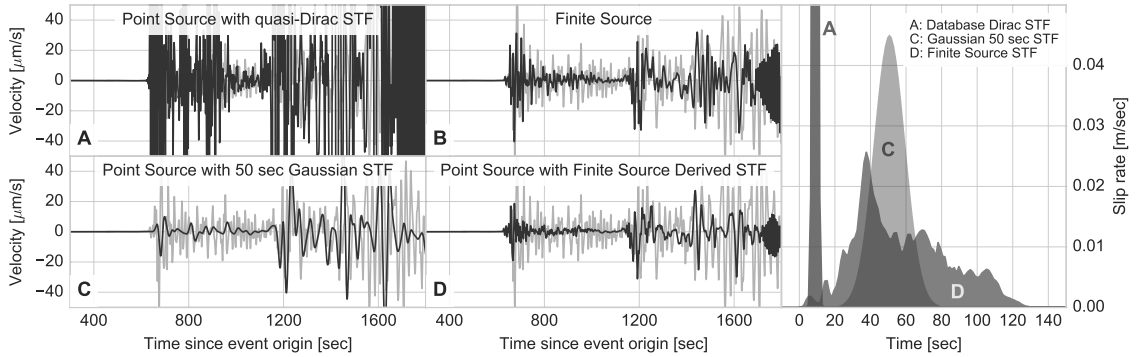

Figure B.3.: Plots of synthetic seismograms for the 2015-04-25 Mw 7.8 Nepal earthquake recorded at the IU.ANMO station. The seismograms are all on the same scale and show (A) a point source calculation with the database native quasi-Dirac delta STF, (B) a calculation using a finite source model from the USGS summing 400 point sources (see the data and resources section), (C) a point source calculation with a smooth Gaussian STF with a width of approximately 50 seconds, and (D) a point source calculation using an STF derived from the finite source model. The gray seismogram is the observed data and the right panel contains the source time functions used.

onset. A source is completely defined by specifying a USGS param file, no other source parameters are required. For the actual calculation of the finite source seismograms we have made some choices that may not be immediately obvious. Upon providing a FFM as a source, the following sequence of events occurs on the server side:

1. The slip rate of each point source is defined as an asymmetric cosine function with a certain rupture, rise, and fall time. We sample it at 10 Hz for a thousand seconds - this limits the maximum rise and fall times. Rise and fall times smaller than one second will be set to one second to make sure it can be accurately sampled. The cosine function is specific to the USGS param file, other FFM file formats might define a separate STF for each point which would be evaluated here.

2. Each sampled slip rate is zero padded with a number of samples at the beginning and the end (the additional time shift is accounted for later). This is done to avoid running into boundary issues with the following filter.

3. A fourth order Butterworth filter is applied twice (forwards and backwards) resulting in a zero phase filter. The corner frequency is the dominant frequency of the database. This assures we don't introduce frequencies in the convolution that we cannot propagate in the numerical simulation.

4. The seismograms for all point sources are calculated, time-shifted, convolved, and stacked.

The resulting seismograms are sent to the users in the same fashion as the point source seismograms. This has a profound impact on the seismograms, see Figure B.3 for plots of seismograms comparing a point source to a finite source.

B.3.6 Green's Functions

Moment tensor inversions are performed by successively modifying the source mechanism to improve the best fit between data and synthetics until it has converged. This requires the generation of a large number of seismograms from a single origin with varying source mechanisms.

A common way of doing this is to reconstruct seismograms as a linear combination of Green's function from fundamental sources. Syngine supports returning Green's functions in the convention introduced by (Minson and Dreger, 2008), which are readily used by SeisComP3 and other software for moment tensor inversion. Seismograms for an arbitrary source mechanism can be calculated by a linear combination of waveforms from these four sources: A vertical strike-slip fault (SS), a vertical dip-slip fault (DS), a dip-slip fault with a dip of 45° (DD), and an explosive source (EP). Formulas for vertical (u_z), radial (u_r), and transverse components (u_t) are then:

$$\begin{aligned} u_z =& M_{tt}\left[\frac{u_{Z,SS}}{2}\cos(2az) - \frac{u_{Z,DD}}{6} + \frac{u_{Z,EP}}{3}\right] + M_{pp}\left[-\frac{u_{Z,SS}}{2}\cos(2az) - \frac{u_{Z,DD}}{6} + \frac{u_{Z,EP}}{3}\right] \\ &+ M_{rr}\left[\frac{u_{Z,DD}}{3} + \frac{u_{Z,EP}}{3}\right] + M_{tp}\left[u_{Z,SS}\sin(2az)\right] + M_{rt}\left[u_{Z,DS}\cos(az)\right] + M_{rp}\left[u_{Z,DS}\sin(az)\right] \end{aligned} \tag{B.3}$$

$$\begin{aligned} u_r =& M_{tt}\left[\frac{u_{R,SS}}{2}\cos(2az) - \frac{u_{R,DD}}{6} + \frac{u_{R,EP}}{3}\right] + M_{pp}\left[-\frac{u_{R,SS}}{2}\cos(2az) - \frac{u_{R,DD}}{6} + \frac{u_{R,EP}}{3}\right] \\ &+ M_{rr}\left[\frac{u_{R,DD}}{3} + \frac{u_{R,EP}}{3}\right] + M_{tp}\left[u_{R,SS}\sin(2az)\right] + M_{rt}\left[u_{R,DS}\cos(az)\right] + M_{rp}\left[u_{R,DS}\sin(az)\right] \end{aligned} \tag{B.4}$$

$$\begin{aligned} u_t =& M_{tt}\left[\frac{u_{T,SS}}{2}\sin(2az)\right] - M_{pp}\left[\frac{u_{T,SS}}{2}\sin(2az)\right] \\ &- M_{tp}\left[u_{T,SS}\cos(2az)\right] + M_{rt}\left[u_{T,DS}\sin(az)\right] - M_{rp}\left[u_{T,DS}\cos(az)\right] \end{aligned} \tag{B.5}$$

where M_{rr}, M_{tt}, M_{pp}, M_{rt}, M_{rp}, and M_{tp} are the moment tensor components, $u_{X,ZZ}$, $u_{X,DS}$, $u_{X,DD}$, and $u_{X,ES}$ seismograms from one of the elementary sources on the X component, and az is the source-receiver azimuth. See (Minson and Dreger, 2008) for a derivation and further explanation.

Syngine's Green's function service returns these elemental source seismograms for an arbitrary source-receiver geometry. Usage is very similar to requesting normal seismograms except that the source-receiver geometry is specified by epicentral distance and source depth. Table B.3 lists all parameters. Note that the Green's functions returned by Syngine are band-limited with the delta peak being replaced by a narrow Gaussian moment rate function as discussed in previous sections.

B.3.7 Meta Information and Documentation

The service will evolve in the future and it offers an interface to query available models and detailed information for each database. This information includes the frequency range within which each model can accurately deliver seismograms, the embedded source slip rate and other information such as the velocity model and the version of AxiSEM used to generate the model. If data is requested as zipped SAC files it will also contain a log file describing the status for each seismogram requested, including errors for those that could not be generated.

The website (`http://ds.iris.edu/ds/products/syngine/`) contains extensive documentation for the service with detailed information about each of the parameters, example queries, usage guides, tutorials, and an interactive URL Builder to aid in constructing custom queries. Furthermore, all databases are offered for download directly from the IRIS DMC. The heaviest users can therefore run their own Instaseis server to extract a very large number of seismograms without overloading the service and avoiding the latency of sending queries over the internet.

Request type parameters		
greensfunction	1	Green's function request type.
Model parameters		
model	prem_a_5s	Velocity model.
Output parameters		
format	miniseed	Output file format: MiniSEED or a ZIP archive of SAC files.
label	Tohoku	Label to be included in file names.
units	velocity	Units: displacement, velocity,
7	or acceleration.	
dt	0.2	Sampling interval in seconds.
kernelwidth	8	Width of the resampling kernel.
Time parameters		
origintime	2010-02-27T06:34:14	Source origin time.
starttime	2010-02-27T06:34:14	Start time of the synthetic traces.
endtime	2010-02-27T06:34:14	End time of the synthetic traces.
Source/Receiver parameters		
sourcedepthinmeters	15000	The source depth in meters.
sourcedistanceindegrees	72	The epicentral distance between source and receiver in degrees computed on the surface of a sphere.

Table B.3.: All available parameters for the Green's function service of Syngine. Please see the website for a detailed and up-to-date explanation of them.

B.4 Available Earth Models

Earth models, no matter if 1-D, 2-D, or 3-D, are constructed upon an inversion of seismic data which requires a number of choices on data, modeling and inversion scheme. Inevitably, different models therefore satisfy different aspects of the inverse problem to a varying degrees. For instance in the spherically symmetric case, the PREM model has been constructed for a reference frequency and includes long period normal modes, whereas the iasp91 model is derived for body waves. Some models may include anisotropic elastic parameters or density, whereas others are isotropic. Even if laterally averaged, some may work better for continental paths, some for oceanic. As such, it is paramount for a generic webservice such as syngine to accommodate the inevitable variety of existent models and therefore waveforms. Syngine can currently calculate seismograms propagating through seven widely used different Earth models. At the time of writing these are ak135f_1s, ak135f_2s, iasp91_2s, prem_a_2s, prem_i_2s, ak135f_5s, prem_a_5s, prem_a_10s, and prem_a_20s. The first part denotes the used Earth model and the final number the smallest period to which synthetics are accurately modelled. _i_ and _a_ denote isotropic and anisotropic variants. ak135-f is an isotropic variant of the ak135 velocity model (Kennett et al., 1995) with the attenuation and density model taken from (Montagner and Kennett, 1996). iasp91 (Kennett and Engdahl, 1991) and PREM (Dziewonski and Anderson, 1981) are other well known Earth models, see figure B.4 for comparison of the models and figure B.5 for seismograms calculated in all the available models. Currently all models are global and continental - we discuss future plans to include other models in section B.7.

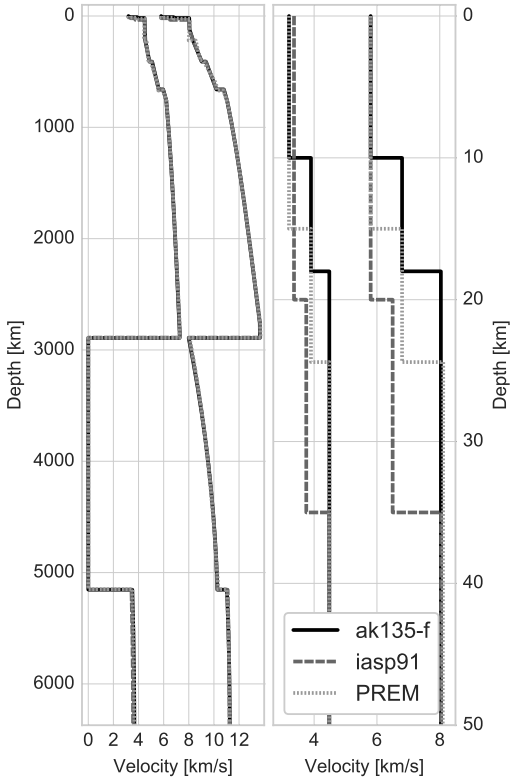

Figure B.4: Plots of the ak135-f, iasp91, and PREM 1-D Earth models for which Syngine can serve seismograms. ak135-f and iasp91 are isotropic models. For PREM, Syngine has an isotropic and an anisotropic variant - only the former is shown here. The left panel shows the full depth range, the right one only the first 50 km as the crust contains most differences. The left curves in each plot are the shear wave velocities, the right ones the compressional wave velocities.

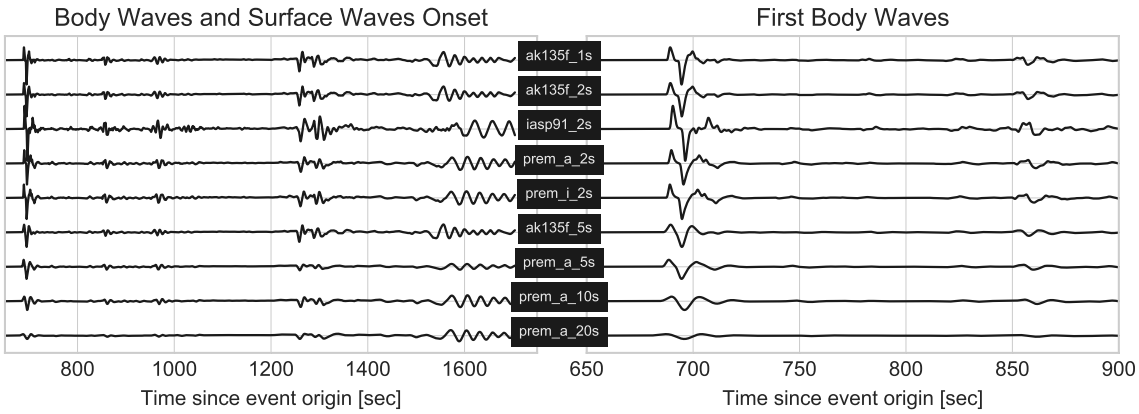

Figure B.5.: Vertical component seismograms for the same source, mechanism, and receiver combination for all models part of the Syngine service at the time of writing. Seismograms from the first arrivals to the onset of the surface waves are shown in the left panel. The right panel zooms in with the first wave package being a mixture of P, pP, sP, and PcP phases and the second at around 855 seconds the PP phase. Black boxes in the middle denote the model from which each seismogram was extracted. Modelled is a Mw 8.3 earthquake near the coast of Chile from 2015-09-16 with a centroid depth of 17.6 km recorded at the IU.ANMO station in Albuquerque, New Mexico, USA with an epicentral distance of approximately 74°. It is computed with a moment tensor source and no further source time function convolution and the smallest resolvable period is ascending from top to bottom.

B.5 Applications

Potential applications for the Syngine service are numerous. We hope that providing an extremely low barrier for access to high-quality synthetic seismograms will spark a number of new uses we have not

yet considered. In general, Syngine is suitable for problems that require the calculation of a reasonable amount of seismograms. The exact number depends on the required frequency content as well as the available internet connection. Studies that require hundreds of thousands of seismograms, for example fully probabilistic source inversions, are better accommodated by Instaseis directly. This section showcases a few potential uses to demonstrate Syngine's applicability to real world problems.

B.5.1 Algorithm Test Bed

Newly developed algorithms have to be tested and benchmarked against synthetic data with fully known and controlled properties. Most wave solvers have not been designed to be used by non-experts and the oftentimes inadequate usability of scientific software (Brown et al., 2015) brings about substantial efforts in order to actually acquire synthetic seismograms. Furthermore, it is very easy to misconfigure software resulting in data that might initially look fine but contains numerous subtle problems. Syngine and the stack it depends on have been thoroughly vetted and benchmarked against established solutions. Additionally, all input parameters are automatically examined and checked for consistency. Data acquired within the constraints presented in this chapter should be accurate. Algorithms that can be validated with Syngine include array processing, phase picking, event locators, back-projectors, and many others.

B.5.2 Data Quality Control

Assessing the quality of recorded data remains a major challenge for network operators and all practicing seismologists handling observed seismograms. Known data problems range from cross-talk amongst recording channels, incorrect sensor orientations to timing errors and erroneous instrument characteristics. Some, like missing or clipped data, are fairly simple to recognize. Others are much harder to detect. Efforts like the IRIS MUSTANG project (`http://service.iris.edu/mustang`) are being pursued with the goal of measuring data characteristics that can be used to quantify data quality. Syngine offers the possibility to augment these quality measurements with comparisons of the observed data to synthetic seismograms where all effects are fully known. Systematic and unexpected deviations from the synthetic seismograms can thus be found and investigated.

An example are small time shifts that are typically introduced by clock errors. The described approach can be applied to various other quality metrics. This is an example demonstrating the feasibility of the approach but it can well be expanded to be performed in a fully-automatic fashion. Figure B.6 visually illustrates the chosen strategy.

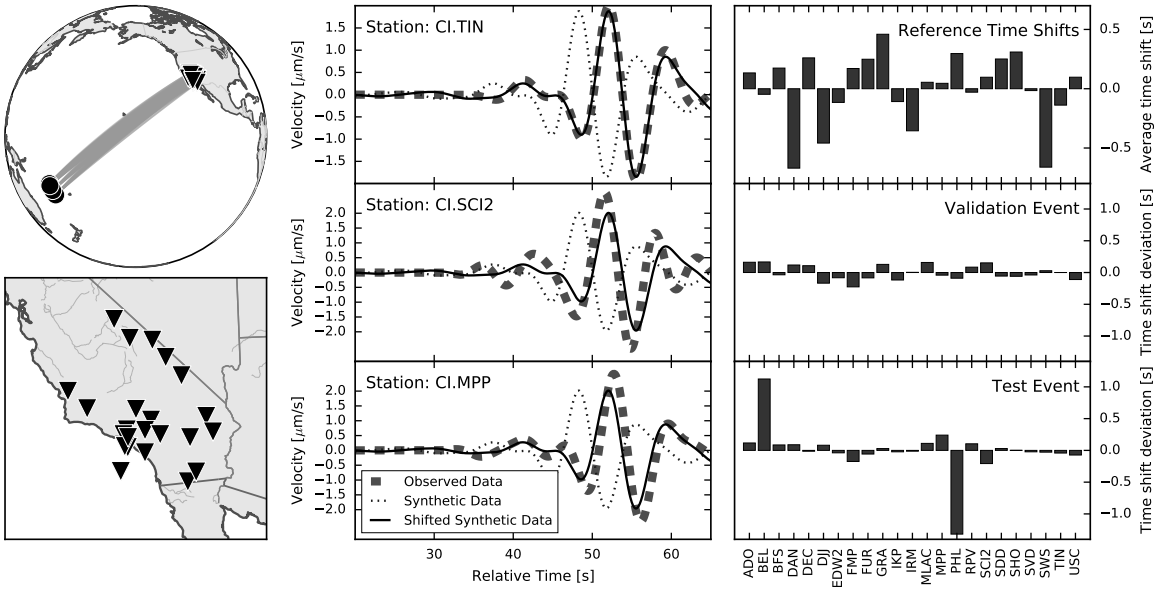

Figure B.6.: Illustration of an application of Syngine to assess the quality of observed data and recording stations. This particular example aims to detect time shifts due to clock errors by cross-correlating observed P phases with synthetic data calculated by Syngine. The left panels show two maps: the top one highlighting the source-receiver geometry, the bottom one zooms in on the receivers, a part of the Caltech Regional Seismic Network (CI). The middle panels show examples of velocity seismograms of observed and synthetic data before and after applying a time-shift equal to the delays measured via cross-correlations. Note that all traces have been processed but no amplitude normalization has been performed - refer to the main text for more details. The rightmost panels are plots of the measured time-shifts. The top one is the reference time shift for each station calculated by using the mean delay from 3 training earthquakes. We subtracted the mean delay of all stations to account for effects such as common unexplained 3-D velocity structure, Earth's ellipticity, and incompatible source parameters. The middle one is the difference in delay time for each station to the reference for an event that was not used to calculate that baseline. The measured shifts are very small. The bottom one finally is another event where we applied an artificial delay of $\pm$ 1 second to two stations which clearly show up in the final measurements. This is only a proof of concept and requires further studies.

We compare observed and synthetic P phase seismograms with a cross-correlation technique yielding time shifts with subsample accuracy (Deichmann and Garcia-Fernandez, 1992). The real data's instrument responses are deconvolved and both synthetic and real data are bandpass filtered with an 8^{th} order zero-phase Butterworth filter to a period band between 5.0 to 12.5 seconds. For each event we subtract the mean time shift from the measured time shifts to account for effects like Earth's ellipticity, unknown and unmodelled 3-D velocity structure, and faulty origin times and depths. These will affect all seismograms approximately equally which is a fact exploited by many techniques, e.g. Waldhauser and Ellsworth (2000). We call this the reference data set and the results are average time shifts per station. Differences amongst these are due to topography and local velocity structure.

Measuring the time shifts for other events and comparing these to the reference data should result in very small differences if the station times are consistent. Otherwise there is some mismatch and likely an error with the time information. Figure B.6 represent both these cases.

These calculations, including the collection of synthetic seismograms from Syngine, can be performed in a matter of seconds making it feasible to be integrated in a continuously running quality control system. The presented approach is a simplified proof of concept but already works quite well. Including more and especially local events, other phases, and more advanced processing and statistics will likely improve the assessment. Note the similarity between observed and synthetic seismograms including matching amplitudes. The proposed technique could thus be extended to check for correct polarities and changing site-effects.

B.5.3 Stability Testing in Source Inversion

The modularity of Syngine allows to quickly test seismological codes versus different velocity models. As an example, we show the result of an inversion for moment tensor and source time function based on waveforms of P-waves. The inversion is based on the method described in Sigloch and Nolet (2006); Stähler et al. (2012). It alternates between updating the moment tensor and the source time function by a joint deconvolution of all Green's functions from the measured P-waveforms. The waveforms are used in a 50 second time window around the P-arrival. The code does a grid-search over all plausible depths (1 to 30 km) and choses the one with lowest misfit. The code is available from `https://github.com/seismology/stfinv`. The inclusion of Instaseis allows it to switch between local waveform databases or the ones provided by Syngine. Since the waveform of the *P-pP-sP* wavetrain, constrains the depth well, but also depends strongly on the velocity model of the crust, different velocity models will result in different estimates for depth and source time function. Inverting the same earthquake with different velocity models provides a qualitative estimate of the stability of the result, which is shown in figure B.7.

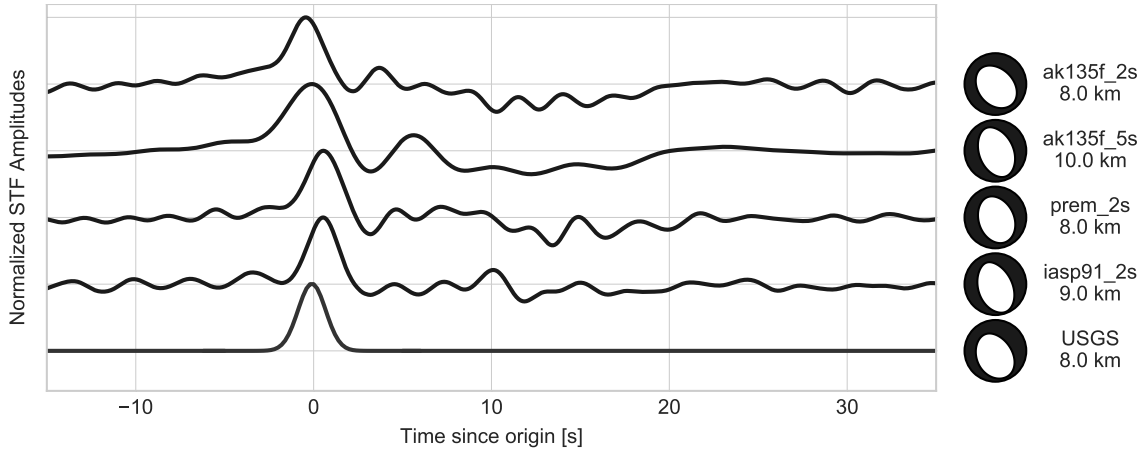

Figure B.7.: Results of a body-wave source inversion for moment tensor and source time function for the Mw 6.2 earthquake in Italy from 2016-08-24. The inversion is performed using databases with four different velocity models: ak135f_2s, ak135f_5s, prem_a_2s, and iasp91_2s. For each model, the depth with the lowest misfit between measured and simulated waveforms is shown. The resulting moment tensors are relatively similar to each other and to the body-wave solution from the USGS (bottom, `quakeml:us.anss.org/focalmechanism/10006g7d/mwr`). The best-fitting depth estimate for the body wave solutions varies between 8 and 10 km. Note that the inversion algorithm does not forbid negative energy in the source time function, if it improves the fit.

B.5.4 Education

Beyond application as a research resource, Syngine serves as a valuable tool for education. Concepts explained by instructors such as polarity of first arrivals, surface wave amplitudes depending on the hypocentral depth, or phase triplications can be visually and interactively discovered and explored by not only by seismologists and undergraduate students. The only requirement is a working internet connection and the capability to read and plot seismograms. Figure B.8 demonstrates two such possibilities. More generally, the intuitive and interactive nature of the tool opens the floor to explore this for school and museum projects.

B.5.5 Backprojection

For large earthquakes, backprojection rupture imaging (Ishii et al., 2005; Trabant et al., 2012) is a useful complement to finite-fault modeling since the rupture velocity often trades off with length in the later ap-

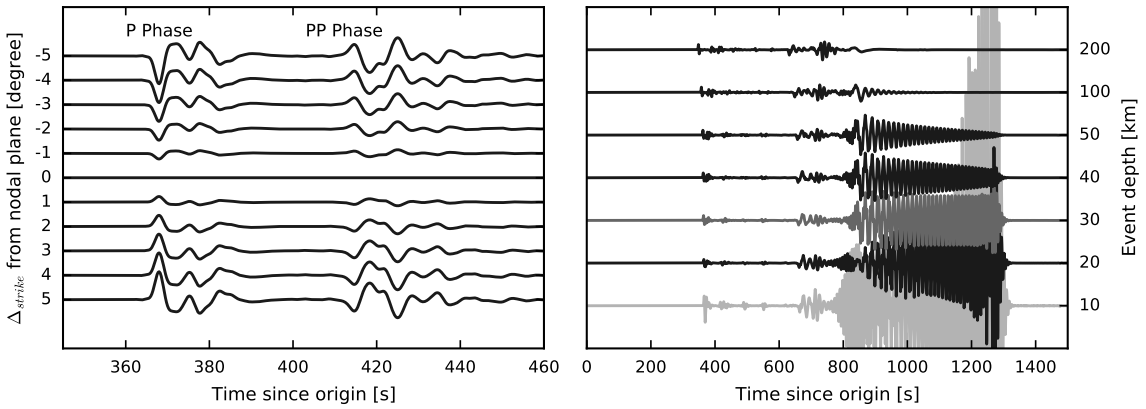

Figure B.8.: Syngine is a very valuable tool to educate future generations of seismologists. Interactively changing source parameters and the geometry of the problem will grant them an enhanced understanding into the behavior of seismograms. This plot demonstrates two examples of this on vertical component seismograms at an epicentral distance of 30 degrees. The left panel shows the effect of nodal planes on seismograms. The source is always the same, except for a differing strike angle. The right panel shows that surface waves are more pronounced for shallower events granting insights into the potential damage of waves originating from different seismic sources.

proach while it can be inferred independently using the former. Figure B.9 highlights Syngine's flexibility and ease of use by comparing backprojection results using P-wave data with equivalent Syngine synthetics using the multi-segment USGS FFM model as a source (`earthquake.usgs.gov/earthquakes/eventpage/usp000g650#finite-fault`). Such a comparison can also provide insight into interpreting backprojection results when they are not clear due to insufficient station geometry or complicated Green's Functions. Downloading the 25 FFM synthetics needed for this example can take as little as three minutes with only a single URL request using the appropriate USGS eventid and network name or a users's station list. This type of exercise can be easily repeated to help guide station configurations for future seismic networks or temporary experiments. Similarly, such an effort can be used to assess resolution issues with perturbations in either the source model or receiver array.

B.6 Discussion

Services such as Syngine that remotely calculate on-the-fly whatever is needed for a particular application are very likely to become more prevalent in the future. Syngine is, to our knowledge, the first service of its kind that grants access to high-quality and customizable synthetic seismograms for Earth with a simple web interface. ETH Zurich recently launched a service in the same vein to generate synthetic seismograms for various proposed Mars models (Ceylan et al., in review). Previously this required high performance computing facilities and considerable technical knowledge, severely limiting its de-facto availability within the research community.

The closest undertaking comparable to Syngine is the ShakeMovie project (Tromp et al., 2010). ShakeMovie produces more physically accurate synthetic data as it includes three dimensional Earth models, topography of the surface, and the effects of gravity and rotation. The major disadvantage is that ShakeMovie can only offer seismograms from a given list of earthquakes and receivers, and it is limited to comparatively low frequencies. Syngine, on the other hand, grants full flexibility in the source-receiver geometry, accomodates frequencies up to 1 Hz, has numerous different models, but is restricted to spherical symmetry. As the 1-D structure explains seismic data, especially body waves, rather well there are a large number of use cases where this limitation is acceptable and the gained flexibility is a worthy trade-

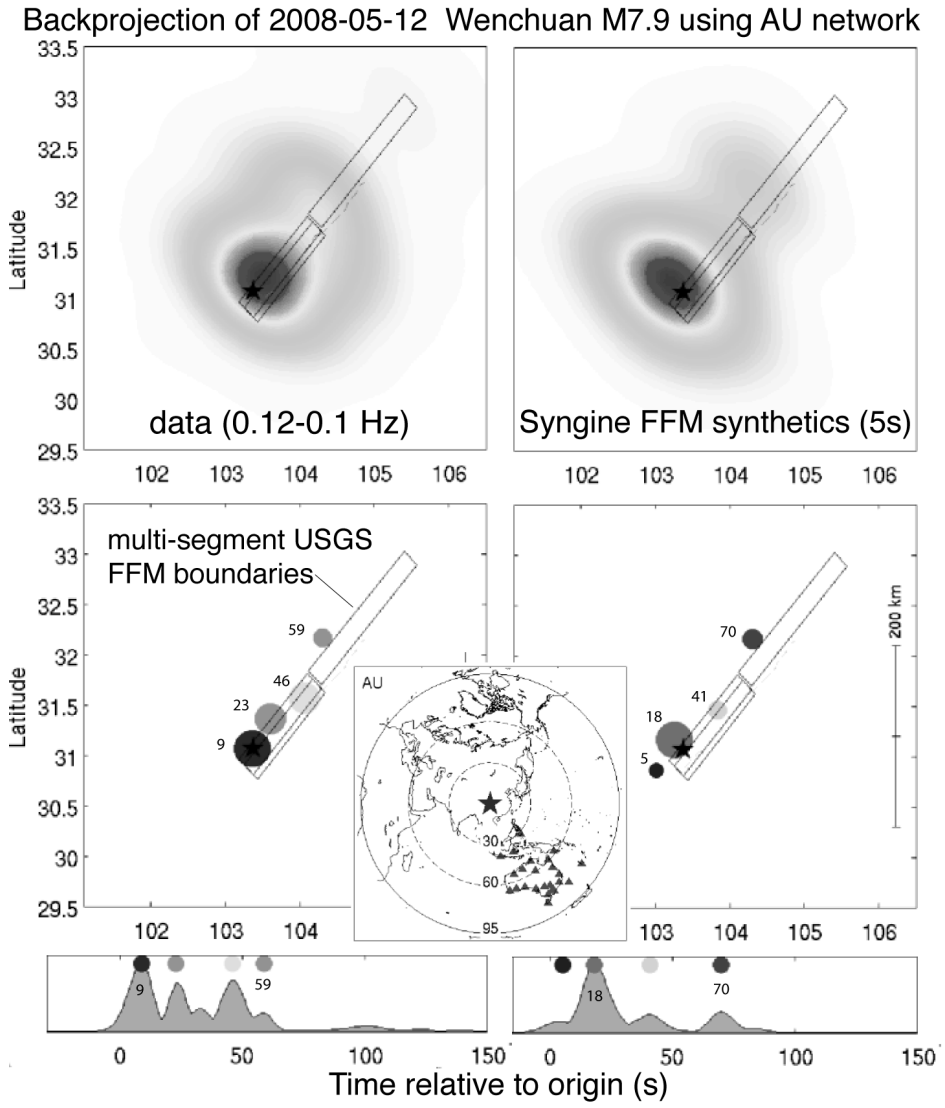

Figure B.9.: Backprojection results for the May 12, 2008 Wenchuan earthquake using data (left) from the AU network (inset) are compared to results using corresponding Syngine synthetics (right) using the USGS multi-segment FFM as input (right). The top panels show the cumulative backprojection energy. The middle panels show the time and locations of local maxima in the backprojection stack amplitude which can be used to infer rupture velocity. While the local maxima in the FFM backprojection appear to be delayed by about 10 seconds relative to the data, this is consistent with the input FFM's moment-rate function (not shown) which peaks at around 20 s after origin time. The bottom panels show peak backprojection stack amplitude over time and appear quite different from each other, however the FFM backprojection amplitude (bottom-right panel) mimics the input FFM moment-rate function to first-order. The FFM is derived using long period body and surface wave data at many azimuths, which may account for this discrepancy. Backprojection amplitudes can be sensitive to small perturbations, but the estimated rupture velocity can be stable when the rupture and source-receiver geometries are favorable. In this example, the inferred rupture velocity from both backprojections is about 2.6 km/s, consistent with the FFM.

off. The combination of capabilities provided by ShakeMovie and Syngine will remain computationally intractable for the foreseeable future.

B.7 Conclusion

We present Syngine, a web service to download on-demand, customized, high-frequency, fully three dimensional seismograms calculated through spherically symmetric, laterally averaged, anisotropic, and viscoelastic Earth models. Source and receiver parameters (locations and mechanism) can be freely chosen as long as the receivers are located at the surface of the Earth. Extraction and calculation of seismograms takes less than a second, granting near instant results.

The present state of Syngine is its first realization and it comes with a number of different global Earth models and seismograms in various frequency bands. Future plans include to add more models, some also with water layers, various lithospheric structures, and regional-specific, even higher frequency models/databases. The web interface is sufficiently generic to accommodate synthetics calculated by other means, for example via normal mode summation if gravity effects are of interest. Syngine is currently an IRIS DMC service but we welcome implementations from other institutions, ideally with compatible interfaces. All software components required to a Syngine web service are openly available.

To increase reproducibility, the code to (re)create all figures except figures B.7 and B.9 in the form of interactive Jupyter notebooks can be found on `http://seismo-live.org`.

Data and Resources

Observed waveform and some used event data can be obtained from the IRIS Data Management Center at `http://www.iris.edu` (last accessed November 2016). This includes waveform data from the Global Seismograph Network (GSN - IRIS/USGS, doi:10.7914/SN/IU), the Southern California Seismic Network (Caltech, doi:10.7914/SN/CI), and the Australian National Seismograph Network. Other earthquake data has been retrieved from the USGS earthquake catalog at `http://earthquake.usgs.gov/fdsnws/event/1/` (last accessed November 2016) and the finite source data for figure B.3 is from the USGS event page at
`https://earthquake.usgs.gov/earthquakes/eventpage/us20002926#finite-fault`
(last accessed October 2016).

Acknowledgements

We thank Editor Zhigang Peng, as well as Mark Panning and an anonymous reviewer for their thoughtful and constructive comments, which helped improve the manuscript. L. Krischer was partially supported by the EU-FP7 VERCE project (number 283543) and also acknowledges support from the EU-funded EPOS project. M. van Driel was supported by grants from the Swiss National Science Foundation (SNF-ANR project 157133 "Seismology on Mars"), S. Stähler by a grant from the Deutsche Forschungsgemeinschaft (project SI1538/2-1 "RHUM-RUM") and T. Nissen-Meyer by European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant 641943. Collaborative visits between some of the authors have been generously supported by eCOST (European Cooperation in Science and Technology) Action ES1401-TIDES. Development and implementation of Syngine at the IRIS DMC was supported by U.S. National Science Foundation awards EAR-1063471 and EAR-1261681.

Acknowlegments

There are many people I have to thank that prepared, enabled, or aided in this thesis, or simply made me have a better time. Many are listed in no particular order - forgive me if I forgot to mention you but make sure to mention it next we meet to see me feel properly ashamed and sorry:

Heiner for opening doors where there were doors and for trusting me enough to do my own thing. For your never-ending enthusiasm that always converted into direct and lasting support of my ventures. And for your encouragement to follow one's dreams, whatever they may be.

Andreas for always making time for me when I needed it. For inviting me several times to Zürich and Utrecht. Thanks for introducing me to your growing group and also for being understanding when you did not hear from me for months at a time as well as enduring my high-tension approaches to meeting deadlines.

Jo for shepherding me through my bachelor and master theses and illuminating the world of observational seismology on many occasions.

Robert and Moritz for introducing me to Python and starting ObsPy together. If I had to choose any singular event that had the largest impact on my professional progression - this would be it.

Tobi for sticking with me through countless hours, days, and nights hacking on ObsPy, preparing releases, and giving workshops.

The whole ObsPy community for being a very skillful motivator in showing that all the things we do are actually used by people worldwide and do make their lifes easier.

The VERCE project and the computational seismology group within EPOS for offering a different point of view on many issues. The umpteen discussions were very draining but valuable.

Jeroen, James, Wenjie, and the computational seismology group in Princeton for showing me how you do things in a similar yet vastly different way. I believe we both gained from the exchanges and especially the developed ASDF data format.

Martin for being great fun in the hot phase of developing Instaseis where our similar points of view but to some extent complementary skill sets were paramount in creating a novel package. Also, I still remember the first time we got the interactive Instaseis GUI working as just a cool and satisfying moment.

Christian B. for explaining numerical optimization theories in the best words and for never making me feel stupid when I asked clearly stupid questions.

Saule, Mike, Laura, Korbinian for discussions of various topics but mainly for making me feel welcome in Zürich.

Kasra for all kinds of tips for thesis writing and layouts and what not and for many discussions on data processing and filtering.

Stephanie, Sia, Chris, Christian P., Alice, Betsy, and others: Thank you not for scientific reasons but for being around to lighten up our at times solitary jobs.

Lorenzo for being a great office mate - usually quite quiet which is good for working, except when not which is also good for breaking the silence.

The administrative team, mainly in Yvonne and Greta, who greatly eased my life through the years by invisibly taking care of many tasks.

Jens for continuous support with all kinds of IT problems I encountered during my run and an always impressively fast response time.

Milena for heroic last-minute proof-reading in a subject so very different from your own.

Last but actually by quite a margin most importantly: Thank you, Lisa, for enabling me to do this. Nothing would have been possible without your never wavering encouragement. A large portion of gratitude also goes to Joschua, Len, and Nolan, for being understanding of my sometimes odd working hours and at times quite frequent trips. All this is for you.