Dear Editor,

In the following paragraphs we reply to the comments of the two referees and the additional comments by Martin van Driel. For each reply we detail what we changed in the manuscript accordingly.

Referee: Tim Greenfield

I.

Comment: Page 2, Line 4: The sentence starting "In physics..." is awkward. I suggest changing into "Physically..."

Thank you for pointing this out. We decided to go for "In the field of physics, ..." to make the meaning clearer without changing it.

Changed: "In the field of physics, ..."

II.

Comment: Page 3, Line 1: the second "let" in the sentence is not required.

Done.

Changed: "... and ~~let~~ f be the force vector ..."

III.

Comment: Page 3, Line 2: I would move the "then" to between "is" and "given"

Done.

Changed: "... ~~then~~ the GF representation is then given by ..."

IV.

Comment: Page 12, Line 9: The sentence starting "Yet..." doesn't sound right. I suggest changing it to "In addition..."

Done.

Changed: "In addition~~Yet~~, seismic targets can provide derivatives of the synthetic displacements ..."

Anonymous Referee #2

Comment: This is a well-written technical paper. The code and the web service look well-established for user applications. But two points are not clear to me.

I.

Comment: Model assumption, 1D/2D/3D? In Section 2.2, the authors present the theory for layered models, for which we can take advantage of the four radiation patterns of the wave solution, i.e., eq.8, the same as those in Nissen-Meyer et al. 2007. First, the correctness of Section 2.2 does not necessarily require the model to be layered; it only requires the model to be axisymmetric with respect to the source, that is, $V_S(r, \theta, \phi) = V_S(r, \theta)$, sometimes referred to as 2D in-plane models. A layered model is a stronger assumption: $V_S(r, \theta, \phi) = V_S(r)$, or a 1D model. It is not clear if their code is designed only for 1D layered models or it can deal with 2D in-plane models or even an arbitrary 3D model. The authors mentioned Crust 2.0, which is inherently a 3D model. On their website, I see they have computed GF's for 1D profiles of Crust 2.0.

We thank the referee for pointing out the need for clarifications.

Pyrocko-GF can handle both: the axisymmetric special case (store types A and B) and a case suitable for 3D heterogeneous media based on the concept of a (densely) gridded source volume and fixed receiver positions (store type C).

On the issue with axisymmetric vs. layered media: While we clearly state in Section 2 that the functions build up on cylindrical symmetry (P6, L13 and L23), we put a focus on layered media due to their more versatile applicability. Nevertheless, the architecture of Pyrocko-GF does not hinder the user to create GF stores for axisymmetric media. Only care has to be taken that such a GF store can only be valid for a fixed epicenter (or fixed receiver location), limiting its use to special problem geometries (e.g. sources below the center of a volcano).

The computation backends currently supported by the Pyrocko-GF's fomosto tool, QSEIS, QSSP, PSGRN, and PSCMP only handle layered velocity models at the moment. Therefore, at the moment, custom import of the GF traces is required for GFs from axisymmetric or 3D media modellers.

Changes

We have adjusted the text in several places to make these issues more clear to the reader.

Section 2.2. Corrected statement: "The number of independent GF components can be further reduced for ~~one dimensional layered~~ media of cylindrical symmetry..."'

Section 3.1. Second paragraph we specify in the second and third sentences: "The physics of the modelled process ~~determines~~ and the symmetry of the medium determine the number of GF components ...", and "For example, in a medium of cylindrical symmetry, with a planar layered medium , and at short distances, the simulation of a full moment tensor requires 10 GF components, ...", respectively.

Section 3.3. Second paragraph, we added: As described above, the specific combination of GF components is defined by the source type and the observed type of quantity (e.g. full moment tensor or a single force, generating far field waveforms or surface displacements, see Tab. 1)..

In the caption of Tab. 1 we clarify: "Configuration type (C) is used for three-dimensionally heterogeneous media with sources in a ~~3D volume box~~ defined source volume and receivers at fixed positions."

II.

Comment: Does the code support both spherical and Cartesian geometry (for local)?

Yes, in our framework (Pyrocko) we use for the definition of location coordinates that are generally given with five numbers. These are geographical coordinates and depths as well as a local Cartesian offset in horizontal east and north direction. For example, a group of locations (stations or sources) can have the same latitude and longitude values but different east shifts, north shifts and depths. This is useful for small scale networks and/or seismicity. Locations across the globe may have latitudes and longitudes with zero Cartesian offsets in east and north. The framework offers many functions to derive distances and azimuths on a sphere and distances are generally distances on spherical surfaces. The coordinate definition and the functions ease the use of GF calculation methods that require either geographical coordinates for calculations with spherical models (e.g. QSSP) or Cartesian input for flat-earth methods (e.g. mode in QSEIS). The GF architecture is in general independent of the coordinate systems. The stores are sorted grids of functions, which the user configures. To make this clearer we complemented the text with some changes and additions.

Changes

In the third paragraph of Section 3.1 after the first sentence we added: "The 'config' file also contains the medium model definitions, i.e. seismic velocities, densities, attenuation coefficients etc. Through selection of a component scheme and configuration type, it defines the mapping used to transform physical source/receiver coordinates into the file lookup indices used in the index file."

In Section 3.5 the first paragraph is expanded with: "... `Source` and `Target` locations are specified as geographic coordinates with an optional Cartesian offset. This design allows the user to handle global, local, as well as mixed setups. These different setups are achieved, respectively, by either setting the Cartesian offset to zero, by setting the geographic coordinates to a common reference location, or by using a combination of both."

III.

Comment: Pressure source in a fluid ocean? ... The model assumptions and requirements for the code should be made explicit in the abstract and conclusions.

We are not sure if we understand this point entirely, because we nowhere mention a fluid ocean. We think the referee is asking if Pyrocko-GF provides a forward modelling scheme for a pressure source in a fluid ocean. Yes, it does. A fluid ocean has a P-wave velocity changing with depth, but zero S-wave velocities. This is easily defined in the ‘config‘ file of a GF store. Using further a component scheme ‘elastic2‘ for an isotropic moment tensor and a store type B would do the trick for the forward calculation of P-Waves at different depth. The store could be filled with GFs computed using the QSEIS or QSSP backend. We introduce the concept and abilities in Sections 3.1 and 3.2. The model assumptions are therefore defined mainly by the GF calculation method, the backend, employed. Otherwise, we point out in the paper that Pyrocko-GF is explicitly open for many different medium and source models with different assumptions and requirements. In the paper we introduce the conceptual architecture for storage. This is why we do not limit the applicability to specific model assumptions and requirements up front so much. However, with this comment we see the need for some more clarifications. Specifically, to clarify the potential use of Pyrocko-GF and the model assumptions for the ready-to-use software solutions to which we refer to as "the Pyrocko framework".

Changes

In the abstract, line 10, we changed: "The framework ~~integrates~~ aids in the creation of such GF stores by interfacing a suite of established numerical forward-modelling codes in seismology ~~, and can incorporate new user-specified GF calculation methods.~~ (computational backends). So far interfaces to backends for layered Earth model cases are provided, however, the architecture of Pyrocko-GF is designed to cover backends for other geometries (e.g. full 3D heterogeneous media) and other physical quantities (e.g. gravity, pressure, tilt). Therefore, Pyrocko-GF defines an extensible GF storage format..."

In Section 3.2, we added more information about the model assumptions of the GF calculation methods: "These GF calculation methods assume horizontally layered medium models, apart from AxiSEM, which requires axisymmetric heterogeneity only."

In the conclusions, we changed: "The command-line interface tool *fomosto* within the Pyrocko-GF framework manages the generation of GFs through the ~~backends~~ computational backends, which so far exist for layered media. *fomosto* also facilitates the comparison, visualisation and quality check of stored GFs. The software is open source and encourages the contribution of use-specific extensions and adding interfaces to other computational backends."

IV.

Comment: comparison with Instaseis/SyngineAbout Instaseis/Syngine, the authors wrote "However, these rigid database schemes are restricted to the modelling method that has been used to create them, and they are confined to specific moment tensor applications." "specific moment tensor" may not be correct. As a user of Instaseis/Syngine, so far as I know, Instaseis and Syngine accept an arbitrary moment tensor (based on eq.8) and Instaseis can also handle point forces (for receiver-wise reciprocity database).

We thank the referee for that comment and we apologize for that incorrect statement. Indeed instaseis/syngine can simulate other sources as well.

Changes

Paragraph 4. Introduction, last sentence, we corrected "However, ~~these~~ rigid database schemes ~~are~~ can be restricted to the modelling method ~~that~~ which has been used to create them~~, and they are confined to specific moment tensor applications~~."

V.

Comment: Because their contribution has a similar purpose and follows similar principles to Instaseis/Syngine, its advantage and generalisation should be correctly and clearly explained. Two advantages seem clear to me: an Instaseis database can be generated only by AxiSEM, while a Pyrocko database is compatible with any forward simulation methods, as claimed by the authors; besides, Instaseis/Syngine is only for global scale.

We have corrected a few unfortunate misrepresentations of Instaseis (see above and below). It generally can support other GF calculation methods than AxiSEM in the future, as stated below by Van Driel. Also, Instasais can be applied regionally. We see the Pyrocko-GF software and the pyrocko framework as a useful complementary code to Instaseis/Syngine. We support other GF calculation methods. We offer the same and some more source types compared to Instaseis, e.g. a direct support four building finite sources, but this may evolve fast in the near future. Whichever code is better suited depends on the problems the user wants to tackle. A detailed comparison is beyond the scope of the presentation and could be outdated soon. In this regard, we added some numbers on the performance to give the user a better feeling on what computational costs may be expected for some setups.

Changes

Section 3.4. Added paragraph: "The computational effort to create a GF store depends on the complexity of the medium model, the temporal and spatial sampling and the duration of the desired waveforms. For example, a global database based on the PREM model, calculated with QSSP, with 2 s sampling and 4 km spatial spacing in distance and source depth, requires an effort of 19 h on a 100-core Intel Xeon E7-8890 high-performance computer and uses 52 GB of disk space. For comparison, regional GF stores at 2 Hz maximum frequency are built within hours on modern desktop computers.

Comments by Martin Van Driel

I.

Comment: There is a paper about IRIS's syngine service, not just a website, so it should be referenced: Krischer, L., Hutko, A., van Driel, M., Stähler, S., Bahavar, M., Trabant, and Nissen-Meyer, T. (2017). On-demand custom broadband synthetic seismograms, Seismol. Res. Lett. 88, no. 4, https://doi.org/10.1785/0220160210.

Thank you. We added this missing reference.

Changed: "... for seismological applications, like e. g. Instaseis (Van Driel et al., 2015) or Syngine (IRIS DMC, 2015; Krischer et al., 2017)."

II.

Comment: P8, L13: "The program INSTASEIS (Van Driel et al., 2015) is suited to calculate approximative wave solutions for a 3D Earth structure with radial symmetry" - This statement is completely wrong, instaseis (not fortran, hence no capitalization in the name), does not solve the wave equation and is hence also not limited to radial symmetry. We use AxiSEM to compute the databases, which is where the limitation to spherical symmetry comes from, but that is completely independent of instaseis, which could easily include any other database source

We apologize for this incorrect statement. We changed the description accordingly.

Changed: "~~The program INSTASEIS (Van Dliel et al., 2015) is suited to calculate approximative wave solutions for a 3D Earth structure with radial symmetry.~~ The program AxiSEM (Nissen-Meyer et al., 2014), as employed by Instaseis (Van Driel et al., 2015), provides seismic wavefields for a 3D axisymmetric Earth structure."

III.

Comment: One of my main concerns when I wrote instaseis was to be accurate enough in space to place receivers anywhere and still get the phase correct so you can apply array methods. Similarly, we wanted to be accurate in the presence of discontinuities. This is in fact a non-trivial interpolation, and we approach it using the spectral element basis, which we discus in some length in our paper. The statement on P10 L2 makes me very suspicious that this issue is treated appropriately here.

We thank Martin Van Driel for the comment. We are aware of the importance of interpolation effects and the need for appropriately fine GF component grid spacing for sufficient accuracy in space and time. We agree that the reader should be made aware of it more explicitly and provide advise. We added text to that effect in Section 3.3 "Source Design" and explain the influence of the user defined grid spacing and interpolation with an expanded paragraph and an additional figure.

Changed:

Section 3.3, from second paragraph on: "The calculation of an observed quantity of interest (e. g. seismic waveforms) for a point source with delta-force excitation at a particular source-receiver constellation is given in section 2. As described above, the specific combination of GF components is defined by the source type and the observed type of quantity (e. g. a full moment tensor or a single force, generating far-field waveforms or surface displacements, see Tab. 1).

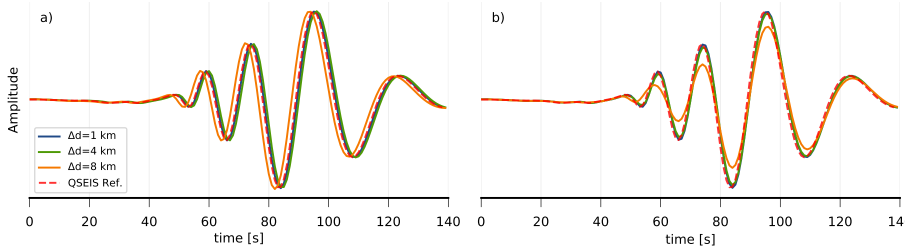

Figure 1: Comparison of interpolation artifacts on synthetic waveforms with different GF grid spacings (a: nearest-neighbour, b: multi-linear). Shown are vertical component displacements, based on GF stores with 1 km, 4 km and 8 km spatial grid spacing (blue, green and orange, respectively) against a reference for the exact source-receiver distance (QSEIS; red). The sampling rate is 2 Hz and the signal contains information up to close to the Nyquist frequency (1 Hz). The waveforms are filtered with a pass band from 0.05 Hz to 0.1 Hz, after interpolation. The medium is layered with important discontinuities of upper crust, lower crust and mantle at 20 km and 35 km depth, respectively. The slowest seismic velocity in the medium is 3.5 km/s. The waveform is simulated for a 10 km deep moment tensor source at a distance of 553.3 km.

The observed quantity at the receiver is a linearly weighted combination of the spatially closest GF components. Often, this requires interpolation to match the requested source-receiver configuration. The interpolation ~~between neighbouring grid nodes can be simple~~ of GF components requires an appropriate density of grid nodes to result in seismograms that are accurate in amplitude and phase (Fig. 1). Furthermore, simulated seismograms vary for different interpolation methods such as nearest-neighbour or multi-linear ~~interpolations. As described above,~~ interpolation. For standard applications with multi-linear interpolation, we could require that the grid spacing d_{grid} should be less than a quarter of the ~~specific combination of GF components is defined by the source type and the observed type of quantity (e. g. a full moment tensor or a single force, generating far-field waveforms or surface~~

~~displacements , see Tab. 1)~~ minimum wavelength. It can be estimated using the minimum wave velocity v_{min} and the maximum signal frequency of the GF traces f_{max} with $d_{\mathrm{grid}} = v_{\mathrm{min}}/(4f_{\mathrm{max}})$. For applications requiring higher accuracy, a smaller grid spacing must be used. For static displacements in the near-field of finite sources discussed below, an appropriate gird spacing is smaller than half the minimum source-receiver distance. In general, smaller grid spacing leads to higher accuracy at the cost of forward-modelling performance and larger GF stores. Interpolation of GF components in the spectral domain is superior but computationally more demanding (Gülünay, 2003)."

A Python framework for efficient use of pre-computed Green's functions in seismological and other physical forward and inverse source problems

Sebastian Heimann[1], Hannes Vasyura-Bathke[2, 4], Henriette Sudhaus[3], Marius Paul Isken[3, 1], Marius Kriegerowski[3, 4], Andreas Steinberg[3], and Torsten Dahm[1,4]

[1]GFZ German Research Centre for Geosciences, Potsdam, Germany
[2]Division of Physical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[3]Institute for Geoscience, University of Kiel, Germany
[4]Institute for Earth and Environmental Sciences, University of Potsdam, Germany

Correspondence: Sebastian Heimann (sebastian.heimann@gfz-potsdam.de)

Abstract.

The finite physical source problem is usually studied with the concept of volume and time integrals over Green's functions (GF), representing delta-impulse solutions to the governing partial differential field equations. In seismology, the use of realistic Earth models requires the calculation of numerical or synthetic GFs, as analytical solutions are rarely available.

The computation of such synthetic GFs is computationally and operationally demanding. As a consequence, on-the-fly re-calculation of synthetic GFs in each iteration of an optimisation is time-consuming and impractical. Therefore, pre-calculation and efficient storage of synthetic GFs on a dense grid of source to receiver combinations enables efficient look-up and utilisation of GFs in time critical scenarios. We present a Python-based framework and toolkit - *Pyrocko-GF* - that enables pre-calculation of synthetic GF stores, which are independent of their numerical calculation method and GF transfer function. The framework ~~integrates~~ aids in the creation of such GF stores by interfacing a suite of established numerical forward-modelling codes in seismology ~~, and can incorporate new user-specified GF calculation methods.~~ (computational backends). So far interfaces to backends for layered Earth model cases are provided, however, the architecture of *Pyrocko-GF* is designed to cover backends for other geometries (e.g. full 3D heterogeneous media) and other physical quantities (e.g. gravity, pressure, tilt). Therefore, *Pyrocko-GF* defines an extensible GF storage format suitable for a wide range of GF types, handling especially elasticity- and wave propagation problems. The framework assists with visualisations, quality control and exchange of GF stores, which is supported through an online platform that provides many pre-calculated GF stores for local, regional and global studies. The *Pyrocko-GF* toolkit comes with a well-documented application programming interface (API) for the Python programming language to efficiently facilitate forward modelling of geophysical processes, e. g. synthetic waveforms or static displacements for a wide range of source models.

Contents

1 Introduction

Green's functions (GFs) are used abundantly to represent force excitations of many different processes inside the Earth. A GF source representation is a mathematical concept to synthesise finite spatio-temporal sources. The GF is defined as the force delta-impulse solution of the system of partial differential equations of the field problem (see e. g. Tolstoy, 1973, chapter 7). In the field of physics, GFs obey the causality condition that allows us to formulate the response at the receiver as a function of retarded time relative to the time of force excitation. For a finite duration of force excitation at the source, this leads to a time convolution integral representation of such a finite-duration source. Sources that have a finite spatial extent are realised by summing over weighted, spatially distributed point sources, leading to a volume integral representation of the source. Combining both integrals, an excitation source can be represented in space and time. The concept of GF source representation was introduced by George Green in 1828 (see e. g. Cannell and Lord, 1993), and since then it is used to describe electromagnetic, mechanical and thermal problems, including elasto-static deformation, wave propagation, wave scattering, and fluid and pressure diffusion.

With a combination of point forces we can approximate the wave fields and deformation for a variety of different processes. For instance, the wave field and the deformation of the well known double-couple moment tensor representation closely resembles a distribution of dislocations on a rupture plane (e. g. Dahm and Krüger, 2014). Such representations of finite-volume sources often include higher order moments and not only single forces, leading formally to elementary solutions, which are spatial and temporal derivatives of the original GFs. In practice, often the nomenclature is not distinguished between the elementary solutions and the point-force GF, and all is denoted as *Green's functions*.

Solving the non-linear inverse problem of determining the parameters of sources with finite duration and finite extent is a computational costly effort. It involves the repeated numerical calculation of a large number of GFs for combinations of force excitations in the context of an optimization algorithm. During the iterations of an optimization the parameters of the force excitations change, e. g. times, locations, orientations and strength. If done on-the-fly, the GF calculation may strongly dominate the overall computational costs of the optimization and even render it impractical.

The structured storage of GFs in *GF databases* (*GF stores* in the following) allows to query pre-calculated GFs for individual source-receiver configurations, rather than time-consuming re-calculation. The conceptional architecture of this approach is sketched in Figure 1. Pre-calculated GFs have been used since long time for operational routine seismological source inversion (Dahm and Krüger, 2014, Tab. 3). Often, these are in-house developed database solutions, or other structured storage, that are linked to specific forward modelling and inversion codes. Some recent developments provide open GF databases for seismological applications, like e. g. Instaseis (Van Driel et al., 2015) or Syngine ~~(IRIS DMC, 2015)~~(IRIS DMC, 2015; Krischer et al., 2017) . The benefit for users is significant, because the expensive calculation of standard GF databases, particularly for global seismic velocity models, ~~has not to be done repeatedly. Also~~ can be evaded. Also making errors in the GF calculation can be avoided by using such approved GF databases. However, ~~these~~ rigid database schemes ~~are~~ can be restricted to the modelling method ~~that~~ which has been used to create them~~, and they are confined to specific moment tensor applications~~.

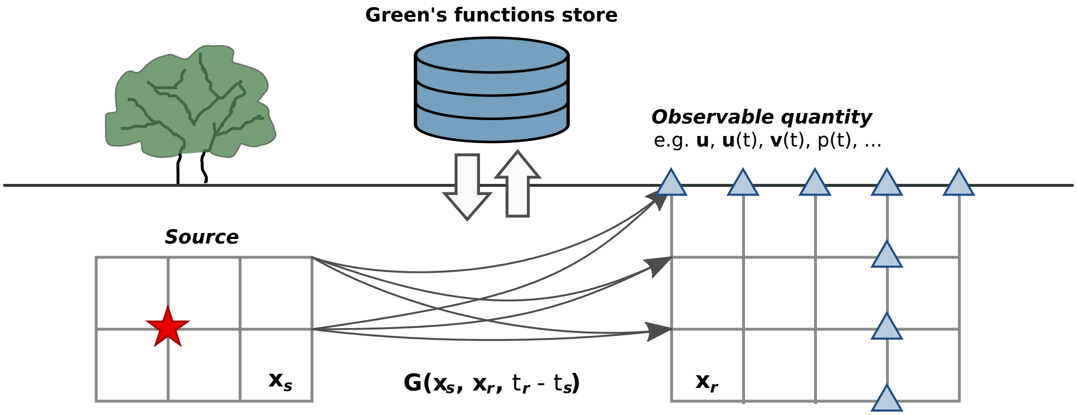

Figure 1. Concept of the Green's Function stores, the positions of the source $\boldsymbol{x}_s$ and the positions of receivers $\boldsymbol{x}_r$ form source-receiver pairs of Green's Functions $G(\boldsymbol{x}_s, \boldsymbol{x}_r, t_r - t_s)$ that are computed on a pre-defined grid for a specified time window. These GFs are saved in a store and can be utilised through the *Pyrocko-GF* Python API.

The aim of our work is to provide a stand-alone open-source toolbox for the calculation and storage of GFs, suited for an easy integration into individual routines. To achieve this, we made our framework independent of the GF type and the GF calculation method that synthesises the physical quantities. The architecture of the GF stores is flexible to allow the storage of extra attributes, e. g. travel time tables. We put special efforts to find a good balance between stability, numerical performance and ~~and~~ user-friendly implementation. While being being part of the *Pyrocko* software library, our framework is easily linked to other seismological Python toolboxes, e. g. *Pyrocko* (Heimann et al., 2017) or *ObsPy* (Krischer et al., 2015). Similarly to the seismological projects Instaseis and Syngine, we also openly share various existing GF stores on https://greens-mill.pyrocko.org.

The paper is organised as follows: section 2 describes the theoretical framework and GF definitions used in the proposed *Pyrocko-GF* infrastructure. Section 3 refers to the implementation design and section 4 gives examples of seismological applications.

2 Theoretical framework

2.1 Basic approach

For the following notations generally it applies that (1) vectors, tensors and matrices in symbolic notations are shown as bold-faced symbols, (2) index notations are expressed with the Einstein summation rule and (3) coordinates refer either, to Cartesian coordinate systems with x-y-z or north-east-down (~~n-e-d~~N-E-D), or to an axisymmetric polar coordinate system with radial-tangential-vertical (r-φ-z, with φ oriented clock-wise and z oriented down). We denote a Green's function by $\mathbf{G} = \mathbf{G}(\boldsymbol{x}_r, \boldsymbol{x}_s, t_r - t_s)$, where $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$ are receiver and source point positions, respectively. The time difference $t_r - t_s$ is the difference between the receiver time t_r and the time t_s of the delta-pulse force excitation at the source. We assume vector forces at the source and vector fields as observed variables at the receiver position. The GF is represented by a second-order tensor, as the three components of the observed vector can be excited by three forces acting in three different directions. Let $\boldsymbol{u}$ be the observed displacement vector field at the receiver located at the surface and, ~~let~~ $\boldsymbol{f}$ be the force vector density acting in the source volume V; ~~then,~~ the GF representation is then given by

$$\boldsymbol{u}(\boldsymbol{x}_r,t_r) = \iiint\limits_V \mathrm{d}^3\boldsymbol{x}_s \int\limits_{-\infty}^{t_r} \mathrm{d}t_s\ \mathbf{G}(\boldsymbol{x}_r,\boldsymbol{x}_s,t_r-t_s)\,\boldsymbol{f}(\boldsymbol{x}_s,t_s) = \iiint\limits_V \{\mathbf{G}(\boldsymbol{x}_r,\boldsymbol{x}_s)*\boldsymbol{f}(\boldsymbol{x}_s)\}\ \mathrm{d}^3\boldsymbol{x}_s. \tag{1}$$

The observed quantity is a superposition of all forces acting in the source volume, convolved with the individual timing of force excitation. In the second GF representation (Eq. 1) the convolution integral is replaced by the symbol $*$, omitting the time variable t. The convolution in Eq. (1) can be rewritten as a matrix equation in any coordinate system convenient for numerical implementation, with three-dimensional data vector $\boldsymbol{u}$ and source vector $\boldsymbol{f}$, and a 3x3 matrix $\mathbf{C}$ of GF components as

$$\mathbf{G}(\boldsymbol{x}_r,\boldsymbol{x}_s)*\boldsymbol{f}(\boldsymbol{x}_s) = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} * \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}. \tag{2}$$

While Eq. (1) is valid in general, a buried seismic source can also be represented by generalised force couples $\mathbf{m}$, localised on a planar surface with an area A (rupture plane). The generalised force couples $\mathbf{m}$ form a symmetric, second-order moment tensor density. In such a moment tensor representation, the GF tensor has 27 instead of nine components. It relates to the first and second-order spatial derivatives of the single-force GF (denoted by a comma-separated last index), as (e. g. Aki and Richards, 2002)

$$u_i(\boldsymbol{x}_r,t_r) = \iint\limits_A \{G^i_{j,k}(\boldsymbol{x}_r,\boldsymbol{x}_s)*m_{jk}(\boldsymbol{x}_s)\}\ \mathrm{d}A \qquad \boldsymbol{x}_s \in A, \tag{3}$$

with $i,j,k \in \{x,y,z\}$ in Cartesian coordinates. In $G^i_{j,k}$, the superscript index i is the displacement direction, while the subscripts j & k indicate the force direction and force arm, respectively.

A matrix formulation of the index-notated Eq. 3 is obtained by mapping the symmetric moment tensor $\mathbf{m}$ into a source vector of length six. Accordingly, the Green's tensor becomes a 3 x 6 matrix:

$$\{G^i_{j,k}(\boldsymbol{x}_r,\boldsymbol{x}_s)*m_{jk}(\boldsymbol{x}_s)\} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \end{pmatrix} * \begin{pmatrix} m_{11} \\ m_{22} \\ m_{33} \\ m_{12} \\ m_{13} \\ m_{23} \end{pmatrix}. \tag{4}$$

Due to the symmetry of the moment tensor, the GF matrix in general has only 18 independent components: $C_{i4} = G^i_{1,2} + G^i_{2,1}$, $C_{i5} = G^i_{1,3} + G^i_{3,1}$, and $C_{i6} = G^i_{2,3} + G^i_{3,2}$.

The given matrix formulation is convenient for an efficient numerical implementation in the GF store and it allows for a generalisation of mixed problems. For example, let $\boldsymbol{u}$ in Eq. (3) be a mixed-data vector $\boldsymbol{b}$ observed at the receiver, consisting of ground displacements $\boldsymbol{u}$ and rotational motions $\boldsymbol{\omega}$. Also, let the source vector $\boldsymbol{s}$ be represented by a mixed-type force system, where force couples and single forces act together within the source volume V. Then, the matrix representation of the mixed problem can be written as

$$\boldsymbol{b}(t_r) \quad = \quad \iiint\limits_V \{\mathbf{C} * \boldsymbol{s}\} \, \mathrm{d}^3\boldsymbol{x}_s \tag{5}$$

with

$$\boldsymbol{b} \quad = \quad (u_n, u_e, u_d, \omega_e, \omega_n, \omega_z)^{\mathrm{T}}, \quad \boldsymbol{s} = (m_{nn}, m_{ee}, m_{dd}, m_{ne}, m_{nd}, m_{ed}, f_n, f_e, f_d)^{\mathrm{T}} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} \mathbf{C}^{\mathrm{MT}} & \mathbf{C}^{\mathrm{SF}} \end{pmatrix}.$$

Note that in Eq. (5) $\mathbf{C}$ contains GF components of different units. If the GF components $\mathbf{C}$ are known over the range of potential source and receiver positions as well as over the relevant time lags, the observables due to arbitrary source models can be synthesised using Eq. (5), by approximating the temporal and spatial integrals with sums (e. g. Dahm and Krüger, 2014).

2.2 Green's tensor and source models in layered elastic media

The number of independent GF components can be further reduced for ~~one-dimensional layered~~ media of cylindrical symmetry in a flat-Earth model or, of spherical symmetry in a spherical-Earth model. For a GF tensor that is calculated for a station at zero azimuth we use the variable $^{\mathbf{0}}\mathbf{G}$. Considering geometric symmetries at zero azimuth (north direction), P-wave motion and vertically polarised S-wave (SV) motion can only be excited by the moment tensor components m_{xx}, m_{yy}, m_{zz}, m_{xz}, and m_{zx}. Similarly, horizontally polarised S-wave (SH) motion can only be excited by the moment tensor components m_{xy} and m_{yx}. This reduces $^{\mathbf{0}}\mathbf{G}$ to the form (Dahm, 1996; Heimann, 2011; Dahm and Krüger, 2014):

$$^{\mathbf{0}}\mathbf{G}^{\mathbf{x}} = \begin{pmatrix} {}^0G^x_{x,x} & 0 & {}^0G^x_{x,z} \\ 0 & {}^0G^x_{y,y} & 0 \\ {}^0G^x_{z,x} & 0 & {}^0G^x_{z,z} \end{pmatrix}, \; ^{\mathbf{0}}\mathbf{G}^{\mathbf{y}} = \begin{pmatrix} 0 & {}^0G^y_{x,y} & 0 \\ {}^0G^y_{y,x} & 0 & {}^0G^y_{y,z} \\ 0 & {}^0G^y_{z,y} & 0 \end{pmatrix}, \; ^{\mathbf{0}}\mathbf{G}^{\mathbf{z}} = \begin{pmatrix} {}^0G^z_{x,x} & 0 & {}^0G^z_{x,z} \\ 0 & {}^0G^z_{y,y} & 0 \\ {}^0G^z_{z,x} & 0 & {}^0G^z_{z,z} \end{pmatrix}. \tag{6}$$

This reduces the number of independent GF components to ten (e. g. Mueller, 1985). These GF components depend only on the source depth, the distance to the receiver and the receiver depth.

A layered medium is invariant to rotation around z at the source element. We exploit this cylindrical symmetry to derive GFs in r-φ-z direction at a station with a given azimuth, through a tensor rotation with the azimuth angle φ:

$$G^{i'}_{p,q} = R_{pj}(\varphi)\, R_{qk}(\varphi)\, {}^0G^i_{j,k} \quad \text{or} \quad \mathbf{G}^{\mathbf{i}'} = \mathbf{R}(\varphi)\; ^{\mathbf{0}}\mathbf{G}^{\mathbf{i}}\; \mathbf{R}^{\mathbf{T}}(\varphi) \quad \text{with} \quad \mathbf{R}(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{7}$$

Note that index i' is in the r-φ-z coordinate system, while i is in the local x-y-z coordinate system at the source (equivalent to north-east-down). Thus, with Eq. (7), the convolution of $\mathbf{m}$ with $\mathbf{G}$ (Eq. 3) is derived component-wise as

$$\begin{aligned}
u_r = \left[G^r_{j,k} * m_{jk}\right]_r &= g_1^r * \left(m_{xx}\cos^2\varphi + m_{yy}\sin^2\varphi + m_{xy}\sin 2\varphi\right) \quad + g_2^r * \left(m_{xz}\cos\varphi + m_{yz}\sin\varphi\right) \\
&\quad + g_3^r * m_{zz} \quad + g_4^r * \left(m_{xx}\sin^2\varphi + m_{yy}\cos^2\varphi - m_{xy}\sin 2\varphi\right) \ , \\
u_z = \left[G^z_{j,k} * m_{jk}\right]_z &= g_1^z * \left(m_{xx}\cos^2\varphi + m_{yy}\sin^2\varphi + m_{xy}\sin 2\varphi\right) \quad + g_2^z * \left(m_{xz}\cos\varphi + m_{yz}\sin\varphi\right) \\
&\quad + g_3^z * m_{zz} \quad + g_4^z * \left(m_{xx}\sin^2\varphi + m_{yy}\cos^2\varphi - m_{xy}\sin 2\varphi\right) \ , \\
u_\varphi = \left[G^\varphi_{j,k} * m_{jk}\right]_\varphi &= g_1^\varphi * \left(\frac{1}{2}\left(m_{yy} - m_{xx}\right)\sin 2\varphi + m_{xy}\cos 2\varphi\right) \quad + g_2^\varphi * \left(m_{yz}\cos\varphi - m_{xz}\sin\varphi\right) \ ,
\end{aligned} \tag{8}$$

with

$$\begin{aligned}
&g_1^r =^0 G^r_{x,x} \ , && g_2^r =^0 G^r_{x,z} +^0 G^r_{z,x} \ , && g_3^r =^0 G^r_{z,z}, && g_4^r =^0 G^r_{y,y} \ , \\
&g_1^z =^0 G^z_{x,x} \ , && g_2^z =^0 G^z_{x,z} +^0 G^z_{z,x} \ , && g_3^z =^0 G^z_{z,z} \ , && g_4^z =^0 G^z_{y,y}, \\
&g_1^\varphi =^0 G^\varphi_{x,y} +^0 G^\varphi_{y,x} \ , && g_2^\varphi =^0 G^\varphi_{y,z} +^0 G^\varphi_{z,y} \ .
\end{aligned}$$

The calculation of synthetic displacements is a linear combination of ten GF components g_j^i, here elementary seismograms, with the moment tensor components as weighting factors. These GF components g_j^i are placed into a GF store. The GF components g_4^r and g_4^z are near-field terms and do not add to regional or teleseismic observations of dynamic displacements (Aki and Richards, 2002).

The GF representation from Eq. (8) can be re-ordered in the form of the matrix notation (Eq. 5)

$$\begin{aligned}
\mathbf{b} &= \begin{pmatrix} u_r & u_\varphi & u_z \end{pmatrix}^T, \quad \mathbf{s} = \begin{pmatrix} m_{xx} & m_{yy} & m_{zz} & m_{xy} & m_{xz} & m_{yz} \end{pmatrix}^T \\
\mathbf{C} &= \begin{pmatrix}
\left[g_1^r\cos^2\varphi + g_4^r\sin^2\varphi\right] & \left[g_1^r\sin^2\varphi + g_4^r\cos^2\varphi\right] & g_3^r & \left[g_1^r - g_4^r\right]\sin 2\varphi & g_2^r\cos\varphi & g_2^r\sin\varphi \\
-g_1^\varphi\frac{\sin 2\varphi}{2} & g_1^\varphi\frac{\sin 2\varphi}{2} & 0 & g_1^\varphi\cos 2\varphi & -g_2^\varphi\sin\varphi & g_2^\varphi\cos\varphi \\
\left[g_1^z\cos^2\varphi + g_4^z\sin^2\varphi\right] & \left[g_1^z\sin^2\varphi + g_4^z\cos^2\varphi\right] & g_3^z & \left[g_1^z - g_4^z\right]\sin 2\varphi & g_2^z\cos\varphi & g_2^z\sin\varphi
\end{pmatrix},
\end{aligned}$$

where $\mathbf{b}$ is the observed data vector (here components of displacements), $\mathbf{s}$ is the source vector (here components of moment tensors), and $\mathbf{C}$ is the matrix form of the GF components. This form is suited for a mixed-type implementation (Eq. 5) and it also allows for a simple projection of $\{\mathbf{C} * \boldsymbol{s}\}$ *a priori* to components of the data vector. These projections can be useful for example, to project to the ray system (L-Q-T), to inclined borehole sensors, or to transform the r-φ-z coordinate system to a ~~n-e-d~~ N-E-D coordinate system of the sensor.

3 Implementation and design

3.1 Green's function store architecture, schemes and storage format

Green's function stores consist of a discrete, sparse and truncated representation of the g_j^i in Eq. (8). Time-dependent 1D arrays (GF traces) are stored for a finite grid of source and receiver positions, or of relative coordinates, resembling a finite spatial extent (Fig. 1). The GF traces are stored in a single binary file `traces`, which allows for efficient reading of the stored data. ~~Lookup~~ Look-up indices for accessing the GF traces are stored in the binary file `index` (for technical details, see Heimann, 2019). Start and end times of the GF traces may be different for every node to allow compact storage of application-relevant portions of the waveforms (e. g. P or S phases, surface waves, etc.). A repeating end-point condition is assumed when extracting waveforms from the store. This way, a step function can be represented by just two samples at every node (e. g. for the modelling of static displacement changes). The strategy we follow to build synthetic data vectors for arbitrary source-receiver locations by using the discretised GF is described below in section 3.2.

We categorise several store configuration types that are tailored to specific groups of source-medium-receiver configurations. The physics of the modelled process ~~determines~~ and the symmetry of the medium determine the number of GF components that are needed to represent the effects of a source at a receiver site (Tab. 1). For example, in a medium of cylindrical symmetry, with a planar layered medium, and at short distances, the simulation of a full moment tensor requires 10 GF components, a single force or isotropic source only needs 5 or 2 components, respectively. Also, for receivers located in the far-field, the number of ~~neceessary~~ necessary components reduces. The combination of these categories defines the structure of the store (Tab. 1). Such a flexible design also allows to store GFs of other transfer functions, e.g. to simulate the poroelastic behaviour of a medium. This categorisation allows to provide efficient calculation, storage and reading of GFs for problems of lower ~~dimension~~dimensions. Consequently, the generation of synthetic ~~quantitites~~quantities, e.g. seismograms, is efficient.

Table 1. Green's tensor component schemes and configuration types for GF stores. Configuration types A and B stand for rotation-symmetric media with (A) common-depth receivers or (B) variable-depth receivers. Configuration type (C) is used for three-dimensionally heterogeneous media with sources in a ~~3D~~ defined source volume ~~(box)~~ and receivers at fixed positions.

Component scheme	Configuration type	Source type	Receiver distances
`elastic10`	A, B	moment tensor	all
`elastic8`	A, B	moment tensor	far-field only
`elastic5`	A, B	single force	all
`elastic2`	A, B	isotropic source	all
`poroelastic10`	A, B	flow rate, pressure	near field (quasi-static)
`elastic18`	C	moment tensor	all

The range and spacing of source locations, depths and lateral source-receiver distances, along with the time span and the sampling rate of the GF traces and other meta data are defined in a configuration file, `config`. The `config` file also contains the medium model definitions, i. e. seismic velocities, densities, attenuation coefficients, etc. Through selection of a component scheme and configuration type, it defines the mapping used to transform physical source/receiver coordinates into the file lookup indices used in the `index` file.

The `config` is defined in the human-readable structured YAML-format. This eases the inspection of the meta data and the file can easily be edited with any text editor. The configuration file can be created and edited either manually or automatically by using the API. For details on the binary encoding and the YAML data structure, please see ~~Heimann (2019)~~ Heimann (2019, Pyrocko GF store file format).

In addition to the mentioned files `traces`, `index` and `config`, the store comprises two directories, `extra` and `phases`. They contain the individual GF method-related configuration file(s) and, in case of waveform problems, the travel time tables for user-specified arrivals. Providing travel time tables consistent with the GFs is useful in practical applications, e.g. to extract specific seismic phase arrivals.

Once created, the grid of traces in a GF store may be decimated in space and time. Such a downsampled version may be placed in an individual store or, in the sub-directory `decimated` of the parent GF store. A downsampled store can be read faster and synthetic quantities are calculated faster. This can be useful, e.g. to simultaneously simulate high frequency body waves and long period surface waves by using the full resolution store and the downsampled store, respectively.

3.2 Green's function computational backend modules

The Green's functions depend on the Earth model and the problem geometry. As scientific applications differ, several specialised numerical methods have been developed to calculate GFs. For instance, reflectivity-type wavenumber integration methods (e. g. QSEIS, Wang, 1999) are commonly used to calculate high-frequency body waves and surface waves from teleseismic to local distances. Direct integration (e. g. GEMINI, Friederich and Dalkolmo, 1995) or hybrid methods (e. g. QSSP, Wang et al., 2017) are established for long-period global seismology on a spherical Earth. QSSP can also be used to simulate the coupling between solid earth, ocean and atmosphere, including gravity waves and infrasound, or to calculate the elasto-gravitational effect from elastic wave propagation. The program ~~INSTASEIS (Van Driel et al., 2015)is suited to calculate approximative wave solutions~~ AxiSEM (Nissen-Meyer et al., 2014), as employed by Instaseis (Van Driel et al., 2015), provides seismic wavefields for a 3D ~~Earth structure with radial symmetry~~axisymmetric Earth structure. Layered elasto-static and gravity problems in seismology and volcanology are often approached with methods adopted from Haskell-type integration (e. g. PSGRN/PSCMP, Wang et al., 2006). Poroelastic quasistatic problems of fully coupled deformation and flow in layered media can be handled using orthonormal Haskell-type integration (e. g. POEL, Wang and Kümpel, 2003). These GF calculation methods assume horizontally layered medium models, apart from AxiSEM, which requires axisymmetric heterogeneity only.

Different GF computation methods are implemented with different model parameter conventions (e. g. for the different coordinate systems, source model descriptions, etc.). They are also written in different programming languages (e. g. Fortran or C). Therefore, we designed the structure of the GF store to be independent from the calculation method of the GFs. A standardised set-up and configuration is ensured by so-called *backends*, written as Python modules. These backends transfer the output of the respective GF computation method to the data-format of the GF store. Such a concept enables the implementation of various existing or future GF calculation methods. The structured filling process of the store allows parallel computation of GFs to efficiently exploit the computer systems at hand.

At the time of publication we have implemented backends for QSEIS, QSSP, PSGRN/PSCMP and POEL to be installed individually and to be used with the Python toolbox *Pyrocko* (Fig. 2a, and Heimann et al., 2017). In addition to these external code backends, *Pyrocko* includes its own method AHFULLGREEN. This backend calculates GFs for an analytical, homogeneous full space (Aki and Richards, 2002) including near-, intermediate- and far-field wave components. It can be applied to simulate acoustic emissions in mines or laboratory probes, or it can be useful for lecturing purposes. The chosen structure is not limited to GFs of force excitation. Also, GFs simulating other processes could be handled through GF stores, e. g. flow rate and pressure pulses. Only the implementation of a method-related backend is needed here.

3.3 Source design

To simulate the physical quantity related to points of force excitations (sources), several GF traces from the pre-calculated store are combined. In the following, we illustrate the concept of source design based on a rectangular dislocation source that is finite in space and time, with uniform slip, a rupture nucleation point and a rupture velocity.

The calculation of an observed quantity of interest (e. g. seismic waveforms) for a point source with delta-force excitation at a particular source-receiver constellation is given in section 2. As described above, the specific combination of GF components is defined by the source type and the observed type of quantity (e. g. a full moment tensor or a single force, generating far-field waveforms or surface displacements, see Tab. 1).

The observed quantity at the receiver is a linearly weighted combination of the spatially closest GF components. Often, this requires interpolation to match the requested source-receiver configuration. The interpolation ~~between neighbouring grid nodes can be simple~~ of GF components requires an appropriate density of grid nodes to result in seismograms that are accurate in amplitude and phase (Fig. 3). Furthermore, simulated seismograms vary for different interpolation methods such as nearest-neighbour or multi-linear ~~interpolations. As described above,~~ interpolation. For standard applications with multi-linear interpolation, we could require that the grid spacing d_{grid} should be less than a quarter of the ~~specific combination of GF components is defined by the source type and the observed type of quantity (e. g. a full moment tensor or a single force, generating far-field waveforms or surface displacements , see Tab. 1)~~minimum wavelength. It can be estimated using the minimum wave velocity v_{min} and the maximum signal frequency of the GF traces f_{max} with $d_{\mathrm{grid}} = v_{\mathrm{min}}/(4 f_{\mathrm{max}})$. For applications requiring higher accuracy, a smaller grid spacing must be used. For static displacements in the near-field of finite sources discussed below, an appropriate gird spacing is smaller than half the minimum source-receiver distance. In general, smaller grid spacing leads to higher accuracy at the cost of forward-modelling performance and larger GF stores. Interpolation of GF components in the spectral domain is superior but computationally more demanding (Gülünay, 2003).

To constrain the finite-duration of moment release at a source point, we extend the delta-force excitation to force excitations in time, which is known as the source time function (STF). Common STF types are triangular, half-sinusoidal, boxcar or smooth-ramp (Brüstle and Müller, 1983; Udias et al., 2014). All of these are defined through a source duration parameter

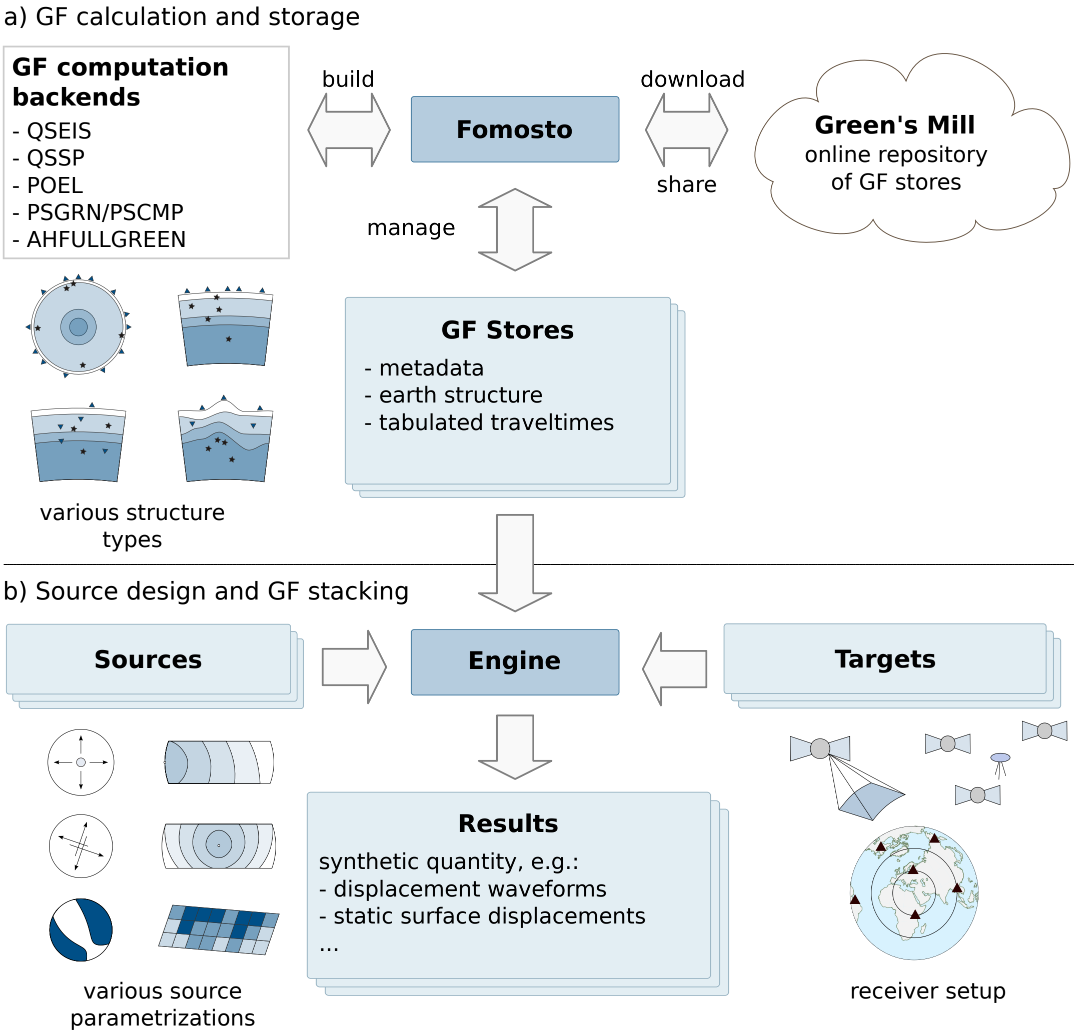

Figure 2. The architecture of *Pyrocko-GF* and object oriented implementation scheme. a) The *fomosto* program is a command-line interface (CLI) to create GFs using the backends, to inspect and manage GF stores. b) The *engine* is the central object that accesses the computed Green's Function stores and calculates the synthetic waveforms based on the specified source (e. g. moment tensor, double-couple source or finite-rectangular) and targets (e. g. virtual seismometers, GNSS stations, or InSAR line-of-sight displacements).

and STF specific parameters, e. g. the peak-moment for a triangular STF. The temporal discretisation of finite-duration source models is corresponding to the temporal sampling of the stored GF traces, Δt_{GF}.

The linear combination of several of these point sources allows to construct spatially finite sources. For example, a rectangular source can be expressed with point sources that are aligned on a rectangular grid outlining the area of the finite source. The source area is defined by the length and width of the rectangular plane, and it's strike and dip. In combination with the

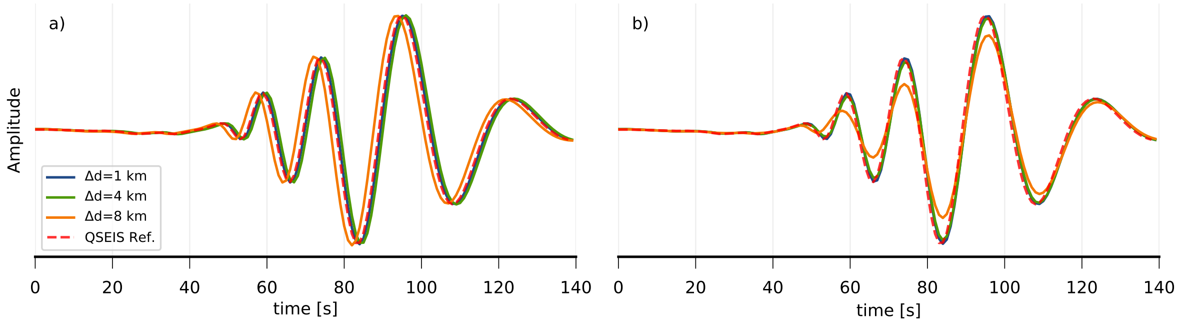

Figure 3. Comparison of interpolation artifacts on synthetic waveforms with different GF grid spacings (a: nearest-neighbour, b: multi-linear). Shown are vertical component displacements, based on GF stores with 1 km, 4 km and 8 km spatial grid spacing (blue, green and orange, respectively) against a reference for the exact source-receiver distance (QSEIS; red). The sampling rate is 2 Hz and the signal contains information up to close to the Nyquist frequency (1 Hz). The waveforms are filtered with a pass band from 0.05 Hz to 0.1 Hz, after interpolation. The medium is layered with important discontinuities of upper crust, lower crust and mantle at 20 km and 35 km depth, respectively. The slowest seismic velocity in the medium is 3.5 km/s. The waveform is simulated for a 10 km deep moment tensor source at a distance of 553.3 km.

rake angle and the amount of slip on the rupture plane, the mechanism of the dislocation and the moment are defined. The nucleation point of the rupture across the fault is defined relative to the center of the rectangular plane.

The spatial distance of point sources on the plane is controlled by the spatial discretisation of the stored GFs (Δz_{GF} and Δx_{GF}). The point source spacing is also constrained by the temporal sampling intervals Δt_{GF}, and the minimum possible rupture velocity $v_{\mathrm{rupt,min}}$. The distance between point sources, Δ_{src}, should either be smaller than half the minimum GF grid spacings or half the distance it takes the rupture to propagate in Δt_{GF}:

$$\Delta_{\mathrm{src}} < \frac{1}{2}\min\left(\Delta z_{\mathrm{GF}}, \Delta x_{\mathrm{GF}}, \Delta t_{\mathrm{GF}} \cdot v_{\mathrm{rupt,\,min}}\right). \tag{9}$$

Accordingly, the integer number of equally-spaced point sources $N_{\mathrm{L,src}}$ along the length of a rectangular, planar source L_{src} is:

$$N_{\mathrm{L,src}} = 1 + 2 \cdot \left\lceil \frac{L_{\mathrm{src}}}{\min(\Delta z_{\mathrm{GF}}, \Delta x_{\mathrm{GF}}, \Delta t_{\mathrm{GF}} \cdot v_{\mathrm{rupt,\,min}})} \right\rceil. \tag{10}$$

The same relations apply for the number of sources across the rectangular plane $N_{\mathrm{W,src}}$ of width W_{src}. Thus, the total number of point sources N_{src} that build the finite source is $N_{\mathrm{src}} = N_{\mathrm{L,src}} \cdot N_{\mathrm{W,src}}$. In case the point sources contribute equally to the moment of the finite source, the point source moment is the total moment divided by the number of point sources N_{src}. In general, the sum of all point source moments is the total moment of the discretised source.

Finally, we define the rupture propagation across the rupture plane by the rupture velocity v_{rupt}. The rupture starts at the nucleation point and from there propagates radially across the fault. The other point sources start rupturing with a time shift of

$t_{\mathrm{shift},i}$ at the *i*th point source. This time shift depends on the distance d_i between the point source and the nucleation point and the mean rupture velocity $\overline{v}_{\mathrm{rupt},\,i}$ between these two points:

$$t_{\mathrm{shift},\,i} = \frac{d_i}{\overline{v}_{\mathrm{rupt},\,i}}. \tag{11}$$

With a similar strategy many more finite-source types can be realised using *Pyrocko-GF* (for details see section 3.5 and the online documentation at https://pyrocko.org/docs).

Once the source model has been discretised into an ensemble of point sources in space and time, a discrete approximation of (1) can be used to calculate a synthetic seismogram or any other observable quantity at a receiver site. The sample $u[i]$ at discrete time i can be computed from a sum of weighted and time-shifted GF contributions as

$$u[i] = \sum_j G[j, i - k[j]]w[j] \tag{12}$$

where $w[j]$ are weighting coefficients and $k[j]$ are delay times. The index j selects the stored GF trace and is a mapping of source position, receiver position, and the GF component index. If it is required to interpolate between the nodes of stored GFs, multiple such contributions from a neighbourhood in GF space are added. Similarly, time interpolation can be implemented by adding weighted and time shifted GF contributions. The weights $w[j]$ are computed considering (1) a rotation at the source (from the source coordinate system to the GF coordinate system), (2) a rotation at the receiver (from the GF coordinate system to the receiver component orientations) and (3) the involved interpolations. Our implementations of source design and GF stacking approach in *Pyrocko-GF* take into account additional practical considerations to provide an efficient, flexible and easy-to-use modelling engine.

3.4 Computation and quality checks of Green's function stores

With the user interface *fomosto* (FOrward MOdeling and Storage TOol) we provide a tool to create, manage and inspect GF stores. *fomosto* builds the GF stores and facilitates the calculation of the GFs through the backends. Furthermore, *fomosto* features chopping of the GF traces to travel-time-relative time windows before their insertion into the GF store. This reduces the amount of stored data and removes synthetic seismogram parts prone to high contamination by numerical noise. Another store-managing feature of *fomosto* is the spatial and temporal decimation of the GF store (Sec. 3.1). Last but not least, *fomosto*-built GF stores can be accessed by the *Pyrocko* project for many forward-modelling applications (see following Sec. 3.5).

The quality control of the ensemble of GF traces is not trivial. GF stores can contain GFs for a large number of source and receiver positions such that a complete visual control by the scientist is impossible. In order to facilitate quality control, *fomosto* offers several options for visual inspection. E. g. in a report generated with its *report* sub-command, the GF traces are assembled in specified time windows, in pre-defined frequency ranges together with predicted travel time arrivals (Fig. 4). The standard report produces plots of displacement and velocity traces, plots of maximum amplitudes and plots of displacement spectra from the GF traces in the analysed store(s). These plots help identifying spurious signals or numerical artefacts in synthetic seismograms. The reports can be customised through a dedicated configuration file. *fomosto* can also be used to visualise the meta data of the GF store, e. g. the Earth model of seismic velocities or the associated travel time tables.

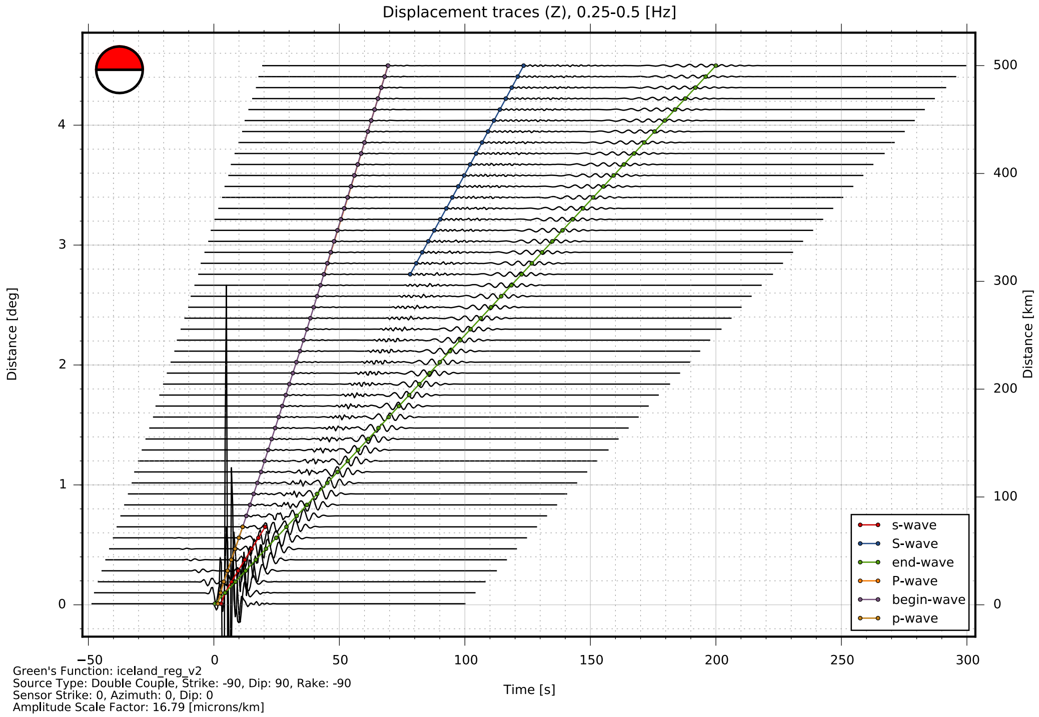

Figure 4. Synthetic seismogram distance section generated by the `report` sub-command of *fomosto* (generated by command `fomosto report single min_config` where *min_config* is a configuration file). Vertical component seismograms are calculated for a given moment tensor source for a given frequency range and distance range provided in GF store (0.25-0.5 Hz at 0-500 km distance). Theoretical first arrivals (P body waves) and other travel time curves are plotted together with synthetic seismograms. Such a display is useful to identify numerical artifacts. Here, an acausal numerical phase running with a negative velocity into negative times at distances below 50 km can be found.

~~Pre-calculated~~ The computational effort to create a GF store depends on the complexity of the medium model, the temporal and spatial sampling and the duration of the desired waveforms. For example, a global database based on the PREM model, calculated with QSSP, with 2 s sampling and 4 km spatial spacing in distance and source depth, requires an effort of 19 h on a 100-core Intel Xeon E7-8890 high-performance computer and uses 52 GB of disk space. For comparison, regional GF stores at 2 Hz maximum frequency are built within hours on modern desktop computers.

To avoid redundant calculations, pre-calculated GF stores are provided for download on the *Green's Mill* web-service under https://greens-mill.pyrocko.org. As of today, GF stores for global and regional applications are provided for a set of different ~~global 1D~~ global Earth models and ~~regional models adapted to the~~ individual layered regional models adapting selected CRUST 2.0 ~~crustal velocity models ()~~velocity profiles (Bassin et al., 2000). GF stores developed during new applications and external projects can be uploaded to the GF store web page *Green's Mill*.

3.5 Forward modelling programming interface

The *Pyrocko-GF* framework implements the described methods to utilise the GF stores for forward modelling of seismic waveforms and near-field displacements. The framework is focused on various forward modelling applications in seismology and geophysics. The object-oriented programming model provides `Source` objects defining the dislocation source(s), `Target` objects defining the modelling target (e. g. seismometer ~~components~~ or GNSS station components), and the `Engine` object being responsible for forward modelling (Fig. 2b). The `Source` objects define the source properties, which may include the location, mechanism and origin time. The `Source` object is responsible for the discretisation of point- and finite-extent sources into their moment tensor representations for weighting of the store's GF traces (see Eq. 8). The `Target` object defines the parameters of the synthesised quantities such as location of receivers and the requested quantity type. ~~Yet~~In addition, seismic targets can provide derivatives of the synthetic displacements (i. e. velocity and acceleration). `Source` and `Target` locations are specified as geographic coordinates with an optional cartesian offset. This design allows the user to handle global, local, as well as mixed setups. These different setups are achieved, respectively, by either setting the cartesian offset to zero, by setting the geographic coordinates to a common reference location, or by using a combination of both.

The `Engine` connects the `Source` and `Target` objects to the GF store. Based on the configuration of ~~the~~ `Source` and `Target`, the `Engine` extracts the required GF traces from the store and realises the stacking of delayed and weighted traces and their subsequent convolution with the `Source`'s STF. Finally, the `Engine` returns the synthetics for the request defined by the ~~`Target` and~~ `Source` and `Target` objects. An example illustrating the the use of the *Pyrocko-GF* API is given in listing 1.

In practice, the described stacking of GF traces is computationally demanding. Stored GF traces are time dependent and differ in length. Consequently, a stacking of these GF traces requires time-shifting of GF traces and dynamic resizing of the output response. GF stores can become large, depending on resolution in time and space. To overcome limitations, we make use of virtual memory mapping of files. Parallelised C extensions manage queries to the database and efficiently stack the GF traces, according to the different component schemes (Section 3.1). An adaptive stacking scheme is available for waveforms and static models, which often have a very large number of source-receiver pairs. To ensure the correct complex interplay of functions software unit tests are carried out routinely.

The presented architecture in *Pyrocko-GF* enables simple implementation of custom point and extended dislocation sources, as well as STFs. The framework's flexibility allows modelling of finite earthquake ruptures with variable STFs across the rupture plane (Vasyura-Bathke et al., 2019). The complete documentation of *Pyrocko-GF*'s API is available online at https://pyrocko.org/docs/current/, featuring tutorials and examples of Python scripts and Jupyter notebooks.

Listing 1. Python script example for forward modelling a synthetic seismogram with the *Pyrocko-GF* framework. More complete examples are available on https://pyrocko.org/docs/current/library/examples/gf_forward.html

```python
from pyrocko import gf, io

engine = gf.LocalEngine(store_superdirs=['.'])
store_id = 'global_2s'
```

```python
# Define three seismogram targets representing East-North-Vertical(up) channels
targets = [
    gf.Target(
        quantity='displacement',
        lat=37.29, lon=-121.31,
        store_id=store_id,
        codes=('', 'STA', '', channel_code))
    for channel_code in 'ENZ']

# Initialise a Double-Couple dislocation source
source_dc = gf.DCSource(lat=52.41, lon=13.06, magnitude=7.3)

# Create synthetic seismograms
response = engine.process(source_dc, targets)
synthetic_traces = response.pyrocko_traces()

# Save seismograms to Mini-SEED files
io.save(synthetic_traces, 'output/%(station)s-%(channel)s.mseed')
```

4 Applications

4.1 Seismic and acoustic source inversion

Challenges in seismic source inversions include computational aspects, as for instance the estimation of parameter uncertainties, and the aspect of a flexible definition of force representations for the specific rupture process under study. Several groups of researchers use GF stores and the *Pyrocko* framework for source parameter estimation and centroid moment tensor inversion. Examples include nuclear explosion source studies (Cesca et al., 2017; Gäbler et al., 2018), anthropogenic triggered events (Sen et al., 2013; Ma et al., 2018; Grigoli et al., 2018), sinkhole processes (Dahm et al., 2011), local and regional tectonic earthquakes (e.g. Dahm et al., 2018), magmatic-induced low-frequency earthquakes (Hensch et al., 2019), earthquakes related to caldera collapses (Gudmundsson et al., 2016; Cesca et al., 2019), landslides (Kulikova et al., 2016), or meteor explosions (Heimann et al., 2013). These examples show that *Pyrocko* GF stores (in some cases its predecessors) have been applied to synthesise data from global, regional and local seismic networks, with different frequency ranges and different wave modes (body waves, surface waves, W-phases, etc.). Also, the range of studied source types is wide including full moment tensors, double-couple, single-force sources, finite-extent earthquake sources and point or finite-extent volcanic sources.

A standard application is to estimate the parameters of a double-couple source from teleseismic data by using a GF store based on a global velocity model. We provide a commented step-by-step code example for the real-data optimisation of the 2009 L'Aquila earthquake (Italy) in form of a Python *Jupyter notebook* at https://github.com/pyrocko/pyrocko-notebooks/blob/master/Waveform_Inversion-Double-Couple.ipynb.

A more exotic example is the source inversion of the Chelyabinsk Meteor's Terminal Explosion (Heimann et al., 2013). It demonstrates the flexibility towards specialised source-types and GF stores involving the atmospheric layers. The Chelyabinsk Meteor's explosion occurred on 15 February 2013 near the town of Chelyabinsk, Russia, at an altitude of about 23 km.

Through seismo-acoustic coupling, its shock-wave generated Rayleigh waves that were observed at epicentral distances of up to 4000 km. We analysed these recordings with a modelling/inversion approach. Synthetic GFs for atmospheric, surface, and underground sources were calculated using the QSSP program (Wang et al., 2017), which can handle the coupling of the solid Earth and atmosphere. The inversion for source location and strength, including error analysis, was performed using the Kiwi Tools (Heimann, 2011), a software for earthquake source inversion. The GF store approach allowed us to cleanly separate the inversion from the GF generation task with only minimal modifications applied to the two distinct code bases. The GF store used for the Chelyabinsk Meteor explosion source inversion is available at https://greens-mill.pyrocko.org/chelyabinsk.

The separation of GF handling, source modelling and inversion/optimisation encourages designing modular software architectures with well-defined interfaces. At the time of writing, two feature-rich, open source earthquake analysis software packages rely on *Pyrocko-GF* for modelling and GF handling: the Bayesian Earthquake Analysis Tool (BEAT, Vasyura-Bathke et al., 2019), which implements Bayesian inference for source characterisation, and the probabilistic earthquake source inversion framework *Grond* (Heimann et al., 2018), which implements a Bayesian bootstrap technique.

4.2 Inverting near-field surface displacements during magma chamber evacuation

In geophysics, static displacement in the near-field is often modelled assuming a homogeneous half-space, using analytical solutions (e. g. Mogi, 1958 or Okada, 1985). Many methods for modelling more realistic dislocation sources and media are available, but at higher computational cost. A GF store approach can compensate this obstacle and provide a convenient access to a more complex modelling environment. Our *fomosto* tool facilitates the computation of static GF stores for layered media with its PSGRN-PSCMP (Wang et al., 2006) backend. To illustrate the use of *Pyrocko-GF* for static deformation modelling, let us examine a case study on the volcano-tectonic crisis in the Comoro Islands, which has started in 2018 offshore Mayotte Island (Cesca et al., 2019).

On Mayotte Island four GNSS stations showed large horizontal displacements and subsidence of up to 20 cm over a period of more than eight months, the distinct horizontal displacements point towards a deflation source situated offshore. We used the few near-field GNSS data to complement the seismic analysis in constraining the location and character of the deflation source. From the seismic analysis they inferred an upward migration of the seismicity from 20 km to shallower crustal levels Cesca et al. (2019).

We tested different point dislocation sources and velocity/elasticity models against the observed GNSS data and evaluated the different models in a Bayesian inversion, provided by the *Grond* framework (Heimann et al., 2018). We found that an isotropic source model in a homogeneous half-space cannot adequately represent the observations. However, the observations are compatible with a constrained moment tensor based point-source model, representative for a vertically elongated, deflating ellipsoid, similar to the point compound dislocation model (Nikkhoo et al., 2016) or an ellipsoidal cavity ~~(Segall, 2010, Chap. 7)~~ (Davis, 1986; Segall, 2010, Chap. 7). Such a tailored dislocation source and more realistic elasticity model derived from the AK135 model[1] (Kennett et al., 1995) provide a more physical representation of the phenomenon and thus a better interpretation of the underlying geological processes.

[1] GF Store https://greens-mill.pyrocko.org/ak135_static-b18151

In this case study, *Pyrocko-GF* provided ~~an~~ a fast and efficient modelling framework, where a ~~freshly~~ newly designed source model could be plugged in without requiring additional implementation work on the forward modelling itself. Seismic waveforms and static displacement are analysed in a homogeneous environment in terms of used Earth models and software toolset.

4.3 Ground motion and shake map simulation

GF stores can be used to explore attenuation functions of ground motions and their uncertainties that are related to the variability of the source and the structural parameters of the medium. Such explorations can be applied in scenario simulations to support risk assessment procedures. Furthermore, it is possible to generate near real-time ground motion forecasts after an earthquake. The *Pyrocko-GF* framework is sufficiently fast to allow for such near real-time waveform simulations. The advantage of synthetic GF-based approaches to empirical relations of peak ground motion (Dahm et al., 2018) and magnitudes (Dahm et al., 2019) is; (1) that regional Earth structure models can be taken into account and (2) that earthquake sources can be simulated in regions where natural earthquakes have not been recorded. The latter situation also arises in most cases of induced seismicity, for which hazard and risk assessment is of particular importance. A challenge in induced and micro-earthquake ground motion simulations is, however, to realistically simulate synthetic seismograms at sufficiently high frequencies.

Dahm et al. (2018) used a synthetic GF approach to validate peak velocity attenuation relations in central Germany, where damaging earthquakes did strike in historical times, but not during the instrumental period. Two similar GF stores were created to calculate peak ground velocity for two deep crustal earthquakes. One store contained GFs for a medium with a hard rock surface layer, while the other store had a soft sedimentary surface layer (find the corresponding GF stores *crust2_m5_hard_top_16Hz* and *crust2_m5_16Hz* at https://greens-mill.pyrocko.org. GFs are pre-calculated for distances between 0 and 300 km, for source depths ranging between 0 and 35 km, at intervals of 1 km with a sampling rate of 16 Hz. Peak ground velocities were extracted for 6 Hz low-pass filtered seismograms ($\mathrm{PGV}_{6\mathrm{Hz}}$). The $\mathrm{PGV}_{6\mathrm{Hz}}$ attenuation reproduced well the observed averages and their variability for the two deep crustal earthquakes, while empirical attenuation relations derived for other regions over or under-predicted the observations. In Fig. 5 we show, as an example, PGV_{6Hz} shake map simulations for a magnitude $\mathrm{M_W}$ 6.3 earthquake scenario near the cities of Leipzig and Halle, Germany[2].

Peak ground motions of displacements or velocities are used to estimate earthquake magnitudes. Usually, empirical attenuation relations are used to normalise measured peak values and to estimate station magnitudes. The empirical relations are often poorly calibrated for the region under study, and only valid for a narrow range of instruments, source depths and strengths. GF stores can overcome some of the limitations of empirical attenuation curves in magnitude scales and can be used to estimate moment magnitudes from peak ground motions. Examples are provided in Dahm et al. (2019) for local magnitude estimations. Figure 6 gives an example how the effect of different crustal models on full waveforms can be simulated on-the-fly[3]. This can be used, for instance, to explore the station variability of global surface wave magnitude scales.

[2] See shake map code example at https://pyrocko.org/docs/current/examples/gf_forward.html

[3] See Rayleigh wave waveform variability code example at https://pyrocko.org/docs/current/examples/gf_forward.html

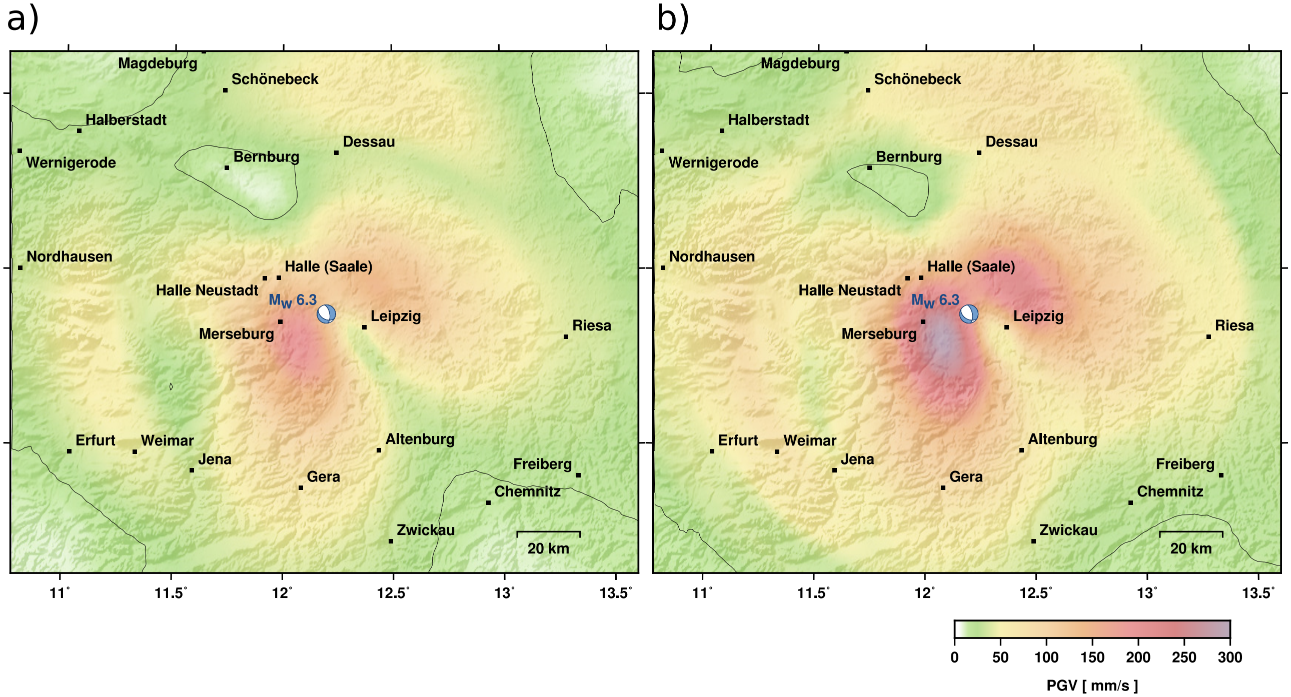

Figure 5. Shake map scenario of a $\mathrm{M_W}6.3$ characteristic lower crust earthquake in 25 km depth between Halle and Leipzig, Germany. (a) Predicted $\mathrm{PGV_{6Hz}}$ assuming hard rock upper layer model. (b) Predicted $\mathrm{PGV_{6Hz}}$ assuming a 500 m thick soft layer beneath the surface with P and S wave velocities of 2.5 km/s and 1.2 km/s, respectively. The contour line shows the 20 mm/s threshold (see Dahm et al., 2018, for further explanation).

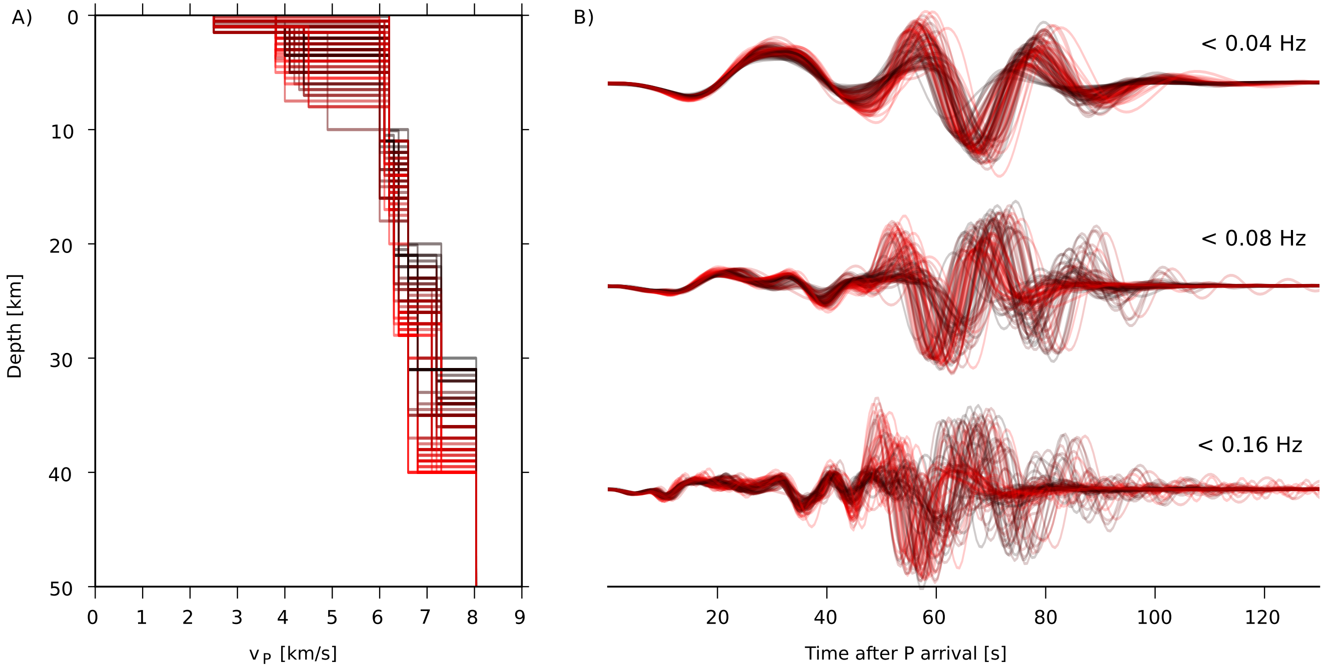

Figure 6. Illustration of waveform sensitivities caused by crustal velocity variations. (a) Colour-coded crustal P-wave velocity profiles as a function of depth. Darker colours correspond to lower crustal thickness. (b) Synthetic, low-pass filtered seismograms of a double-couple point source corresponding to the P-wave velocity structure in (a) at 278 km distance).

4.4 Array and network design optimisation by full waveform simulation

The simulation of full waveforms for the directed Monte-Carlo optimization of seismic networks is computationally demanding and not standard in seismology. López-Comino et al. (2017) used synthetic GF stores for the simulation of full waveforms to predict the network performance and spatial and temporal variation of the magnitude of completeness before the deployment of stations. Synthetic velocity seismograms were generated and modified by real recorded noise based on a randomised catalogue of synthetic earthquake sources for a given source volume in the area under investigation. The seismograms were then used in a full waveform event detector to evaluate the expected magnitude of completeness.

For the design of a small-aperture array for the study of local micro-earthquakes, a synthetic GF store has been used by Karamzadeh Toularoud et al. (2019). For a given distribution of synthetic hypocenters and source mechanisms that simulate an earthquake swarm, the authors randomly generated array locations and station configurations, for which synthetic seismic waveforms were computed. For these synthetic setups the expected beam power and the slowness calculation error was optimised, which allowed the authors to judge on a preferred array installation configuration. The array setup was fixed to use seven stations with three components. 4000 different station configurations have been tested with 100 different sources each. Hence, a total number of 8.4 million traces had to be generated for the analysis. A desktop computer was sufficient for the computational task. Karamzadeh Toularoud et al. (2019) used the GF store https://greens-mill.pyrocko.org/vogtland_malek2004_sed_HR-95ef03.

5 Conclusions

Pyrocko-GF is a Python module that provides functionality to calculate Green's functions (GFs) and to organise these in a store. In *Pyrocko-GF*, backends utilise different numerical calculation methods for GFs, making the stores independent of the actual type of GFs. The command-line interface tool *fomosto* within the *Pyrocko-GF* framework manages the generation of GFs through the ~~backends~~computational backends, which so far exist for layered media. *fomosto* also facilitates the comparison, visualisation and quality check of stored GFs. The software is open source and encourages the contribution of use-specific extensions and adding interfaces to other computational backends.

Our implementations support computation of GFs useful for earthquake-related problems, which include GFs for broadband global, regional and local seismograms, optionally with near-field terms, as well as static surface displacements as measured by GNSS sensors and/or through InSAR. Furthermore *Pyrocko-GF* stores are prepared to store infrasound, oceanic infragravity, poro-elastic GFs and elasto-gravitational GFs. On the GF online platform *Green's Mill* (https://greens-mill.pyrocko.org) we provide GF stores for download that are useful for many standard earthquake-related applications using global velocity models.

The *Pyrocko-GF* framework allows to efficiently access GF stores to synthesise source-specific responses inside the medium for comparison with different observations. Sources as well as source-time-functions can easily be customised without limitations on the source complexity. We discussed seismological applications using *Pyrocko-GF* stores to simulate seismic waveforms, static surface displacements and peak ground motion caused by earthquakes, volcanic activity and exploding meteoroids. The range of applications is not limited to generalised force couples or single forces. With new backends also flow rate, pressure pulse or any other type of excitation and observable, which are representable using the GF approach, could be included in a *Pyrocko-GF* store.

Code and data availability. The presented software is available as open source under the GNU General Public License v3 and can be downloaded from https://github.com/pyrocko. The documentation and examples are available at https://pyrocko.org. Pre-computed Green's function stores can be browsed and downloaded at https://greens-mill.pyrocko.org/.

Author contributions. S.H. and T.D. are the scientific Pyrocko project leader, supported by H.S. and H.V.-B. as scientific project partners. S.H. leads the software implementation for which H.V.-B., M.I. and M.K have contributed significantly and in many ways, e. g. by writing parts of the codes. All authors have contributed in the software design, the code testing and the manuscript writing.

Acknowledgements. H.S., A.S. and M.I. acknowledge funding by the German Research Foundation DFG through an Emmy-Noether Young-Researcher-Grant (276464525).

H.V.-B. was financially supported by King Abdullah University of Science and Technology (KAUST) BAS/1/1353-01-01 as well as by Geo.X, the Research Network for Geosciences in Berlin and Potsdam (Project number: SO_087_GeoX). T.D. and S.H. acknowledge co-funding by SERA (EU Grant 730900). Timothy Willey helped to implement fomosto report plots. University of Rheno, Nevada for processing and providing GNSS deformation data for Mayotte Island (Blewitt et al., 2018).

Competing interests. The authors declare that they have no conflict of interest.

References

Aki, K. and Richards, P. G.: Quantitative Seismology, University Science Books, 2 edn., http://www.worldcat.org/isbn/0935702962, 2002.

Bassin, C., Laske, G., and Masters, G.: The current limits of resolution for surface wave tomography in North America. EOS Trans AGU 81:F897, Eos, 81, F897, 2000.

Blewitt, G., Hammond, W., and Kreemer, C.: Harnessing the GPS data explosion for interdisciplinary science, Eos, 99, 2018.

Brüstle, W. and Müller, G.: Moment and duration of shallow earthquakes from Love-wave modelling for regional distances, Physics of the Earth and Planetary Interiors, 32, 312–324, https://doi.org/10.1016/0031-9201(83)90031-6, 1983.

Cannell, D. M. and Lord, N. J.: George Green, Mathematician and Physicist 1793-1841, The Mathematical Gazette, 77, 26–51, http://www.jstor.org/stable/3619259, 1993.

Cesca, S., Heimann, S., Kriegerowski, M., Saul, J., and Dahm, T.: Moment Tensor Inversion for Nuclear Explosions : What Can We Learn from the 6 January and 9 September 2016 Nuclear Tests, North Korea ?, Seismological Research Letters, 88, https://doi.org/10.1785/0220160139, 2017.

Cesca, S., J., L., Razafindrakoto, H., Rivalta, E., Passarelli, L., Isken, M. P., Petersen, G., Cotton, F., and Dahm, T.: Magmatic intrusion and submarine eruption offshore Mayotte, Comoro Islands. Long period volcano-seismic crises offshore Mayotte, Comoro Islands: indications of submarine caldera collapse, Nature Geoscience, https://doi.org/submitted, 2019.

Dahm, T.: Relative moment tensor inversion based on ray-theory: Theory and synthetic tests, Geophys, J. Int., 124, 245–257, 1996.

Dahm, T. and Krüger: Moment tensor inversion and moment tensor interpretation, in: New Manual of Seismological Observatory Practice (NMSOP-2), edited by Bormann, P. e., pp. DOI: 10.2312/GFZ.NMSOP–2_IS_3.9, IASPEI, GFZ German Research Centre for Geosciences, Potsdam, 2014.

Dahm, T., Heimann, S., and Bialowons, W.: A seismological study of shallow weak earthquakes in the urban area of Hamburg city, Germany, and its possible relation to salt dissolution, Natural Hazards, 0, 10.1007/s11 069–011–9716–9Online First, 2011.

Dahm, T., Heimann, S., Funke, S., Wendt, S., Rappsilber, I., Bindi, D., Plenefisch, T., and Cotton, F.: Seismicity in the block mountains between Halle and Leipzig, Central Germany: centroid moment tensors, ground motion simulation and felt intensities of two $M \approx 3$ earthquakes in 2015 and 2017, Journal of Seismology, https://doi.org/doi: 10.1007/s10950-018-9746-9, 2018.

Dahm, T., Cesca, S., Heimann, S., Maghsoudi, S., and Kühn, D.: A Greenfunction magnitude approach to overcome limitations of traditional magnitude scales, AGU 2017, https://doi.org/Abstract S22C-04, 2019.

Davis, P. M.: Surface deformation due to inflation of an arbitrarily oriented triaxial ellipsoidal cavity in an elastic half-space, with reference to Kilauea volcano, Hawaii, Journal of Geophysical Research: Solid Earth, 91, 7429–7438, https://doi.org/10.1029/JB091iB07p07429, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB091iB07p07429, 1986.

Friederich, W. and Dalkolmo, J.: Complete synthetic seismograms for a spherically symmetric earth by a numerical computation of the Green's function in the frequency domain, Geophysical Journal International, 122, 537–550, 1995.

Gäbler, P., Ceranna, L., Nooshiri, N., Barth, A., Cesca, S., Frei, M., Grünberg, I., Hartmann, G., Koch, K., Pilger, C., Ross, J. O., and Dahm, T.: A Multi-Technology Analysis of the 2017 North Korean Nuclear Test, Solid Earth, https://doi.org/doi.org/10.5194/se-2018-102, 2018.

Grigoli, F., Cesca, S., Rinaldi, A. P., Manconi, A., Clinton, J. F., Westaway, R., Cauzzi, C., Dahm, T., and Wiemer, S.: The November 2017 Mw 5.5 Pohang earthquake: A possible case of induced seismicity in South Korea, Science, 5, 1003–1006, 2018.

Gudmundsson, M. T., Jónsdóttir, K., Hooper, A., Holohan, E. P., Halldórsson, S. A., Ófeigsson, B. G., Cesca, S., Vogfjörd, K. S., Sigmundsson, F., Högnadóttir, T., Einarsson, P., Sigmarsson, O., Jarosch, A. H., Jónasson, K., Magnússon, E., Hreinsdóttir, S., Bagnardi, M., Parks, M. M., Hjörleifsdóttir, V., Pálsson, F., Walter, T. R., Schöpfer, M. P. J., Heimann, S., Reynolds, H. I., Dumont, S., Bali, E., Gudfinnsson, G. H., Dahm, T., Roberts, M. J., Hensch, M., Belart, J. M. C., Spaans, K., Jakobsson, S., Gudmundsson, G. B., Fridriksdóttir, H. M., Drouin, V., Dürig, T., Aðalgeirsdóttir, G., Riishuus, M. S., Pedersen, G. B. M., van Boeckel, T., Oddsson, B., Pfeffer, M. A., Barsotti, S., Bergsson, B., Donovan, A., Burton, M. R., and Aiuppa, A.: Gradual caldera collapse at Bárdarbunga volcano, Iceland, regulated by lateral magma outflow, Science, 353, 1–24, https://doi.org/10.1126/science.aaf8988.sciencemag.org, http://science.sciencemag.org/content/353/6296/aaf8988.abstract, 2016.

Gülünay, N.: Seismic trace interpolation in the Fourier transform domain, Geophysics, 68, 355–369, https://doi.org/10.1190/1.1543221, 2003.

Heimann, S.: A Robust Method To Estimate Kinematic Earthquake Source Parameters, Ph.D. thesis, University of Hamburg, 2011.

Heimann, S.: Pyrocko GF store file format, GFZ Data Services (in preparation), https://pyrocko.org/docs/current/formats/gf_store.html, 2019.

Heimann, S., Gonzalez, A., Wang, R., Cesca, S., and T., D.: Seismic characterization of the Chelyabinsk meteor's terminal explosion, Seismological Research Letters, 84, 1021–1025, 10.1785/0220130 042., 2013.

Heimann, S., Isken, M., Kühn, D., Sudhaus, H., Steinberg, A., Vasyura-Bathke, H., Daout, S., Cesca, S., and Dahm, T.: Grond - A probabilistic earthquake source inversion framework, https://doi.org/10.5880/GFZ.2.1.2018.003, http://pyrocko.org/grond/docs/current/, 2018.

Heimann, Heimann, S., Kriegerowski, M., Isken, M., Cesca, S., Daout, S., Grigoli, F., Juretzek, C., Megies, T., Nooshiri, N., Steinberg, A., Sudhaus, H., Vasyura-Bathke, H., Willey, T., and Dahm, T.: Pyrocko - An open-source seismology toolbox and library., GFZ Data Services, V. 0.3, https://doi.org/doi.org/10.5880/GFZ.2.1.2017.001, pyrocko.org, 2017.

Hensch, M., Dahm, T., Ritter, J., Heimann, S., Schmidt, B., Stange, S., and Lehmann, K.: Deep low-frequency earthquakes reveal ongoing magmatic recharge beneath Laacher See Volcano (Eifel, Germany), Geophys. J. Int., 216, 2025–2036, https://doi.org/doi.org/10.1093/gji/ggy532, 2019.

IRIS DMC: Data Services Products: Syngine, https://doi.org/10.17611/dp/syngine.1, 2015.

Karamzadeh Toularoud, N., Cesca, S., Kr\uger, F., and Dahm, T.: No Title, 2019.

Kennett, B. L., Engdahl, E., and Buland, R.: Constraints on seismic velocities in the Earth from traveltimes, Geophysical Journal International, 122, 108–124, 1995.

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., and Wassermann, J.: ObsPy : a bridge for seismology into the scientific Python ecosystem, COMPUTATIONAL SCIENCE & DISCOVERY, 8, 17, http://dx.doi.org/10.1088/1749-4699/8/1/014003, 2015.

Krischer, L., Hutko, A. R., van Driel, M., Stähler, S., Bahavar, M., Trabant, C., and Nissen-Meyer, T.: On-Demand Custom Broadband Synthetic Seismograms, Seismological Research Letters, 88, 1127–1140, https://doi.org/10.1785/0220160210, 2017.

Kulikova, G., Schurr, B., Krüger, F., Brzoska, E., and Heimann, S.: Source parameters of the Sarez-Pamir earthquake of 1911 February 18, Geophysical Journal International, 205, 1086–1098, 2016.

López-Comino, J., Cesca, S., Kriegerowski, M., Heimann, S., Dahm, T., Mirek, J., and Lasocki, S.: Monitoring performance using synthetic data for induced microseismicity by hydrofracking at the Wysin site (Poland), Geophysical Journal International, 210, 42–55, https://doi.org/10.1093/gji/ggx148, https://academic.oup.com/gji/article-lookup/doi/10.1093/gji/ggx148, 2017.

Ma, J., Dineva, S., Cesca, S., and Heimann, S.: Moment tensor inversion with three-dimensional sensor configuration of mining induced seismicity (Kiruna mine, Sweden), Geophysical Journal International, 213, 2147–2160, https://doi.org/doi.org/10.1093/gji/ggy115, 2018.

Mogi, K.: Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them, Bull. Earthquake Res. Inst. Univ. Tokyo, 36, 99–134, 1958.

Mueller, C. S.: Source pulse enhancement by deconvolution of an empirical Green's function, Geophysical Research Letters, 12, 33–36, https://doi.org/10.1029/GL012i001p00033, http://dx.doi.org/10.1029/GL012i001p00033, 1985.

Nikkhoo, M., Walter, T. R., Lundgren, P. R., and Prats-Iraola, P.: Compound dislocation models (CDMs) for volcano deformation analyses, Geophysical Journal International, p. ggw427, 2016.

Nissen-Meyer, T., van Driel, M., Stähler, S. C., Hosseini, K., Hempel, S., Auer, L., Colombi, A., and Fournier, A.: AxiSEM: broadband 3-D seismic wavefields in axisymmetric media, Solid Earth, 5, 425–445, https://doi.org/10.5194/se-5-425-2014, https://www.solid-earth.net/5/425/2014/, 2014.

Okada, Y.: Surface deformation due to shear and tensile faults in a half-space, Bulletin of the Seismological Society of America, 75, 1135–1154, 1985.

Segall, P.: Earthquake and volcano deformation, Princeton University Press, 2010.

Sen, A., Cesca, S., Bishoff, M., Meier, T., and Dahm, T.: Automated full moment tensor inversion of coal mining induced seismicity, Geophysical Journal International, 195, 1267–1281, https://doi.org/doi.org/10.1093/gji/ggt300, 2013.

Tolstoy, I.: Wave propagation, McGraw-Hill, New York, 1973.

Udias, A., Madariaga, R., and Buforn, E.: Source mechanisms of earthquakes: theory and practice, Cambridge University Press, Cambridge, UK, 2014.

Van Driel, M., Krischer, L., Stähler, S. C., Hosseini, K., and Nissen-Meyer, T.: Instaseis: instant global seismograms based on a broadband waveform database, Solid Earth, 6, 701, 2015.

Vasyura-Bathke, H., Dettmer, J., Steinberg, A., Heimann, S., Isken, M., Zielke, O., Mai, P. M., Sudhaus, H., and Jónsson, S.: (submitted), The Bayesian Earthquake Analysis Tool, Seismological Research Letters, 2019.

Wang, R.: A simple orthonormalization method for stable and efficient computation of Green's functions, Bulletin of the Seismological Society of America, 89, 733–741, 1999.

Wang, R. and Kümpel, H.-J.: Poroelasticity: Efficient modeling of strongly coupled, slow deformation processes in a multilayered half-space, GEOPHYSICS, 68, 705–717, 2003.

Wang, R., Lorenzo-Martín, F., and Roth, F.: PSGRN / PSCMP — a new code for calculating co- and post-seismic deformation , geoid and gravity changes based on the viscoelastic-gravitational dislocation theory, Computers and Geosciences Geosciences, 32, 527–541, https://doi.org/10.1016/j.cageo.2005.08.006, 2006.

Wang, R., Heimann, S., Zhang, Y., Wang, H., and Dahm, T.: Complete synthetic seismograms based on a spherical self-gravitating Earth model with an atmosphere – ocean – mantle – core structure, Geophys, J. Int., 210, 1739–1764, https://doi.org/10.1093/gji/ggx259, 2017.