Dear Anonymous Referee #2,

Thank you very much for reviewing our manuscript. In the following paragraphs we reply each of your comments. For each reply we detail what we changed in the manuscript accordingly.

Kind regards,

Sebastian Heimann

Your review

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.

Thank you 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. If you are 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 can be 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).

Thank you 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. It generally can support other GF calculation methods than AxiSEM in the future, as stated in the comment by Van Driel. Also, Instaseis 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.