Round 1

Reviewer A

Dear author,

This manuscript presents a comparative study of four open-source Python optimization libraries used in 2D time-domain acoustic full-waveform inversion (FWI).

While the title is clear, it may be too broad. The phrase "A Review" implies a more comprehensive comparative study of optimization techniques available in the Python community, whereas only the L-BFGS method is compared in this work. Other local optimization methods, such as conjugate gradient (with various Beta schemes) or Gauss-Newton methods, are not covered, nor are stochastic, Bayesian, or variational inference-based optimization libraries. Narrowing the scope of the study or expanding the comparison to include other methods would enhance the manuscript's focus.

In the introduction, the FWI problem, the benefits of using Python, and the ingredients necessary for FWI (propagator and solver) are well-framed. However, the introduction does not fully address why this review is necessary. For instance, are there gaps in the current literature on comparisons of these Python-based frameworks? Strengthening the motivation for this comparison could be helpful, so as addressing the practical challenges that this comparison seeks to solve.

For the **sotb-wrapper** section, certain technical details, such as the reverse communication protocol, may be difficult to follow for readers unfamiliar with these concepts. This could be especially challenging given that the paper is also intended for undergraduate or graduate students.

The comparison of L-BFGS implementations from the four optimization frameworks—**scipy.optimize**, **sotb-wrapper**, **PyROL**, and **NLopt**—feels insufficient. Additional results regarding the line search methods and performance metrics would be beneficial. Computational efficiency is a key factor that also needs to be mentioned. Yet, none of those are provided. Furthermore, it could be useful to discuss the incorporation of preconditioning operators, such as the functionality provided by the Seiscope Fortran libraries and their Python interface.

The current results presented seem too simple and weak to support the conclusions drawn. For the acoustic FWI case, it is necessary to provide convergence curves such as ||m_fwi – m_true||/||m_true|| or some structural similarity metric in addition to the objective function curves shown in Fig. 3. The number of line search iterations and the specific line search method should also be included to better evaluate the performance of each library.
I suspect the L-BFGS implementation in **scipy** may employ a more progressive line search strategy, which may result in more updates in the final FWI results. I would recommend running all optimizers under similar convergence criteria, such as achieving a 99% decrease in the

objective function value, and then comparing the number of iterations, forward and adjoint propagator evaluations, and other relevant metrics. Given these, the statement that "scipy's L-BFGS-B method demonstrated superior performance" requires further evidence.

In addition to the toy acoustic FWI problem, it would be beneficial to include a case study using **elastic FWI**, where multiparameter inversion challenges could provide further insights. Moreover, applying the frameworks to field data would strengthen the real-world applicability of the comparison.

Finally, since this is a review, the authors might want to propose future directions or open questions. How can these frameworks evolve? How large of a problem can these optimization frameworks handle efficiently? Currently, the manuscript reads more like a summary of available optimization libraries than a full review paper.

Overall, this paper offers an interesting and technical review of Python optimization frameworks for acoustic FWI but requires a deeper analysis when comparing the different libraries, and broader FWI problems more than a 2D acoustic FWI. The writing can also be improved as a potential review article.

Best of luck

Reviewer B

Dear Author,

While I see your manuscript as technically very interesting, I don't see strong novelty. I believe you should emphasize the contribution of the conducted research and express it clearly.

1. The main optimization method that is used for FWI is l-BFGS as it takes 2nd order derivatives into account, but it is not computationally as expensive as BFGS. However, this is not discussed in the Scipy-related section. I encourage the author to focus on this algorithm in the Scipy section and even the other sections.

2. The author should explain what line search methods have been used for each algorithm and each inversion (runs of FWI, Fig. 3.).

3. In section 4, I recommend discussing the reason for choosing l-BFGS, providing examples of other studies using this method or other optimization algorithms.

L31: I don't agree with using the word "quickly" here.

L149: Modify the sentence, "*It also* ***includes*** *includes solvers"*.

L314: Please provide some algorithms (as examples) that are available through NLopt.

L601: The algorithm is called l-BFGS. Please modify the text (lowercase L).

Thank you,

Response to Reviewers

Review for manuscript

Open-source numerical optimization Python libraries for full-waveform inversion: A Review

By

Oscar Mojica

Round 1

Dear Editor,

We sincerely appreciate the time and effort you and the reviewers have dedicated to evaluating our manuscript, "*Open-source numerical optimization Python libraries for full-waveform inversion: A Review*." We are grateful for the constructive feedback, which has been invaluable in refining and strengthening our work. We acknowledge the concerns raised regarding the depth of our review and the need for a more comprehensive comparison of the optimization frameworks, including additional discussions on optimization methods, line search strategies, and computational efficiency.

In response, we have carefully revised the manuscript to incorporate these elements and addressed the reviewers' specific suggestions. Additionally, we have ensured that the manuscript adheres to Seismica's formatting guidelines, as requested.

Below, we provide a detailed point-by-point response to the reviewers' comments, outlining the changes made in the revised manuscript. The reviewers' specific comments are listed in bold text, followed by our responses in normal text.

PS:

- In response to Reviewer A's suggestion, we have modified the title of the manuscript to better reflect the content and scope of the study. The new title is "Open-source gradient-based numerical optimization Python libraries for full-waveform inversion: A Review."
- We have updated Figure 3. The first function evaluation (associated with the starting model) is now paired with a value of 1, which refers to the maximum value of the objective error, as the error is normalized.

Reviewer A:

Dear author,

This manuscript presents a comparative study of four open-source Python optimization libraries used in 2D time-domain acoustic full-waveform inversion (FWI).

While the title is clear, it may be too broad. The phrase "A Review" implies a more comprehensive comparative study of optimization techniques available in the Python community, whereas only the L-BFGS method is compared in this work. Other local optimization methods, such as conjugate gradient (with various Beta schemes) or Gauss-Newton methods, are not covered, nor are stochastic, Bayesian, or variational inference-based optimization libraries. Narrowing the scope of the study or expanding the comparison to include other methods would enhance the manuscript's focus.

Thank you for your thoughtful feedback regarding the title and scope of our manuscript. We appreciate your comments and have taken them into consideration to improve the clarity and focus of our work.

In response, we have modified the title to specifically reference gradient-based optimization techniques. While some of the frameworks discussed include algorithms from other categories (e.g., stochastic), our study focuses on the gradient-based algorithms available in each framework, as these form a common foundation across all the libraries examined. The decision to compare the l-BFGS method was guided by its ubiquitous implementation across the studied frameworks, making it an ideal baseline for comparison. That said, we have listed other optimization algorithms supported by these frameworks to provide readers with a broader context. Additionally, we emphasize that the provided pieces of code in Listings and supplementary material (GitHub repository) were designed to be straightforward and practical. Users can readily adapt the examples to experiment with other algorithms supported by the frameworks. Furthermore, we have included brief comments on the quality and accessibility of the documentation for each framework, which we believe will assist readers in navigating these tools effectively. This can help users explore alternative methods and deepen their understanding.

We hope these changes address your concerns and enhance the manuscript's overall focus and clarity. Thank you again for your valuable insights.

In the introduction, the FWI problem, the benefits of using Python, and the ingredients necessary for FWI (propagator and solver) are well-framed. However, the introduction does not fully address why this review is necessary. For instance, are there gaps in the current literature on comparisons of these Python-based frameworks? Strengthening the motivation for this comparison could be helpful, so as addressing the practical challenges that this comparison seeks to solve.

We appreciate the reviewer's thoughtful feedback. To address this concern, we have added a paragraph in the revised manuscript explicitly highlighting the novelty and significance of our review. While the text addressing this point is concise, we believe it is clear in articulating the significance and relevance of this work.

To the best of our knowledge, this is the first comprehensive comparative analysis of open-source gradient-based optimization frameworks available in Python. As we have emphasized in the revised manuscript, there is currently a lack of such comparative studies, even though they are essential for guiding both researchers and practitioners in choosing the most appropriate tools for their specific needs. Furthermore, our analysis sheds light on potential gaps and limitations in existing frameworks, thus contributing to the broader conversation on how these tools can be improved.

I hope this addresses your concern about emphasizing the novelty and importance of the manuscript.

For the sotb-wrapper section, certain technical details, such as the reverse communication protocol, may be difficult to follow for readers unfamiliar with these concepts. This could be especially challenging given that the paper is also intended for undergraduate or graduate students.

In response, we have included a dedicated paragraph in the revised manuscript to explain the concept of the reverse communication protocol in a clear and accessible manner. With this addition,
we aim to ensure that readers can follow the discussion without difficulty. We hope this change addresses your concern and enhances the accessibility of the manuscript for a broader audience. Thank you again for your insightful comments, which have helped us improve the clarity of our work.

The comparison of L-BFGS implementations from the four optimization frameworks—scipy.optimize, sotb-wrapper, PyROL, and NLopt—feels insufficient. Additional results regarding the line search methods and performance metrics would be beneficial. Computational efficiency is a key factor that also needs to be mentioned. Yet, none of those are provided. Furthermore, it could be useful to discuss the incorporation of preconditioning operators, such as the functionality provided by the Seiscope Fortran libraries and their Python interface.

We appreciate the reviewer's suggestion to expand the discussion on line search methods and computational efficiency. To address this, we have included additional details regarding the line search strategies employed by each framework to provide better insight into their convergence behavior and final results. We have now incorporated a comparative analysis of computational cost based on the number of gradient evaluations required by each l-BFGS implementation.

Regarding preconditioning, we have highlighted this feature in the section discussing the sotb-wrapper framework. Although sotb-wrapper offers a preconditioned l-BFGS implementation, in our experiments, we used the standard l-BFGS version and scaled the gradient by the inverse source-side illumination (source illumination is actually equivalent to the diagonal part of the pseudo-Hessian, which was proposed by (Shin et al., 2001a).). This approach was chosen because source-side illumination can be efficiently computed alongside the gradient and is applicable across all frameworks, as the gradient computation process is largely consistent in the inversion codes used. This strategy ensures an uniform basis for comparison among the different optimization frameworks while maintaining practicality in implementation.

- Shin, C., Jang, S., and Min, D.-J. Improved amplitude preservation for prestack depth migration by inverse scattering theory. Geophysical prospecting, 49(5):592–606, 2001.

The current results presented seem too simple and weak to support the conclusions drawn. For the acoustic FWI case, it is necessary to provide convergence curves such as ||m_fwi – m_true||/||m_true|| or some structural similarity metric in addition to the objective function curves shown in Fig. 3. The number of line search iterations and the specific line search method should also be included to better evaluate the performance of each library.

We would like to clarify the primary objective of this study: our goal is to provide a comparative analysis of the optimization libraries themselves, rather than a detailed evaluation of a specific algorithm. The decision to illustrate the resolution of the FWI using these libraries, with a particular optimization algorithm, was made to demonstrate their applicability. This broader goal guided our choices regarding the data and metrics presented.

Regarding the suggested convergence plots, several practical limitations influenced our approach:

Access to Iterative Solutions: Among the four libraries compared, three provide access to intermediate solutions (m_fwi) at each iteration, while one operates as a "black-box" solver without exposing such details. This inconsistency makes it infeasible to generate uniform convergence curves across all libraries.

Variability in Stopping Criteria: Each library uses different stopping criteria, and there is no single, consistent criterion across all four. For example, while absolute error could serve as a standardized stopping metric, this is not a common option in the libraries we evaluated.

We selected objective function curves as the primary metric because objective function values are a widely available and standard feature in most of the libraries we examined. This choice allows us to compare the libraries based on a commonly accessible parameter, despite the cited restrictions. While the objective function values are available for all libraries, we also encountered inconsistencies in how they are reported. One library does not provide these values per iteration but rather per line search attempt. To maintain a fair comparison, we aligned the number of function evaluations with another library that imposes the same Wolfe conditions in its line search.

We hope this explanation clarifies our approach and the rationale behind the choice of metrics and visualizations in the manuscript. We are happy to consider further suggestions on how to address these limitations while maintaining the overarching goal of our study.

I suspect the L-BFGS implementation in scipy may employ a more progressive line search strategy, which may result in more updates in the final FWI results. I would recommend running all optimizers under similar convergence criteria, such as achieving a 99% decrease in the objective function value, and then comparing the number of iterations, forward and adjoint propagator evaluations, and other relevant metrics. Given these, the

statement that “scipy’s L-BFGS-B method demonstrated superior performance” requires further evidence.

We appreciate your insightful observations regarding the L-BFGS implementation in SciPy and the proposed approach for a more uniform comparison. However, as mentioned earlier, the optimization libraries evaluated in this study each define their own stopping criteria, which vary significantly. This variability makes it unfeasible to enforce a common convergence criterion, such as achieving a 99% decrease in the objective function value, across all libraries.

That said, we recognize the importance of assessing computational effort across different implementations. To address this, we have included the number of gradient evaluations, which serve as a meaningful measure of computational workload. These values provide insight into the effort required by each library under the chosen stopping conditions.

Regarding the statement that "SciPy’s L-BFGS-B method demonstrated superior performance," we acknowledge that this could be misinterpreted as referring to computational efficiency. To clarify, our intended meaning was that SciPy's L-BFGS-B produced final results that were closest to the true model. This outcome is likely due to its use of the strong Wolfe conditions in line search, which may contribute to better convergence in this particular problem. We have revised the wording to avoid any unintended implications regarding computational speed.

In addition to the toy acoustic FWI problem, it would be beneficial to include a case study using elastic FWI, where multiparameter inversion challenges could provide further insights. Moreover, applying the frameworks to field data would strengthen the real-world applicability of the comparison.

We appreciate your suggestion to extend the study to include an elastic FWI case and field data. While we recognize the value of such additional experiments, our primary focus in this work is to provide a comparative analysis of optimization libraries using a well-controlled, reproducible setup. Expanding to multiparameter inversion in elastic FWI would introduce additional complexities that,
while insightful, are beyond the scope of our current study.

However, to facilitate further exploration by the community, we provide our complete experimental setup, including the source code and a containerized environment, allowing readers to easily extend the experiments. Additionally, as our implementation is built on Devito, which supports code generation for various physical models—including elastic wave propagation—adapting the provided framework for elastic FWI should be relatively straightforward, particularly for 2D cases. Devito's official repository already includes examples of elastic wave modeling that can serve as references (https://github.com/devitocodes/devito/blob/master/examples/seismic/tutorials/06_elastic_varying_parameters.ipynb).

For real-world field data applications, additional preprocessing steps and adjustments may be required, but the provided codebase can serve as a solid foundation for such extensions. We hope that by making our implementation fully available, we encourage further investigations into these more complex scenarios.

Finally, since this is a review, the authors might want to propose future directions or open questions. How can these frameworks evolve? How large of a problem can these optimization frameworks handle efficiently? Currently, the manuscript reads more like a summary of available optimization libraries than a full review paper.

We have incorporated a discussion on the evolution and maintainability of the optimization frameworks, highlighting their development trajectories and the factors influencing their continued support. Additionally, we have addressed the scalability of these frameworks, clarifying that while we presented a 2D example for demonstration purposes, these methods can efficiently handle 3D problems and larger-scale optimizations.

Overall, this paper offers an interesting and technical review of Python optimization frameworks for acoustic FWI but requires a deeper analysis when comparing the different libraries, and broader FWI problems more than a 2D acoustic FWI. The writing can also be improved as a potential review article.

By incorporating the discussion on framework evolution, and scalability, we aim to provide a more comprehensive review rather than a simple summary of available libraries. We appreciate this suggestion and have updated the manuscript to reflect these aspects.

Best of luck

Recommendation: Revisions Required

Reviewer B:

Dear Author,

While I see your manuscript as technically very interesting, I don't see strong novelty. I believe you should emphasize the contribution of the conducted research and express it clearly.

Thank you for your thoughtful feedback and for recognizing the technical interest in our manuscript. I appreciate the opportunity to clarify the contribution of our research and how it addresses a critical gap in the literature. In response to your comment, we have added a specific paragraph in the revised manuscript to explicitly highlight this novelty and its importance. While the text addressing this point is concise, we believe it is clear in articulating the significance and relevance of this work.

To the best of our knowledge, this is the first comprehensive comparative analysis of open-source gradient-based optimization frameworks available in Python. As we have emphasized in the revised manuscript, there is currently a lack of such comparative studies, even though they are essential for guiding both researchers and practitioners in choosing the most appropriate tools for their specific needs. Furthermore, our analysis sheds light on potential gaps and limitations in existing frameworks, thus contributing to the broader conversation on how these tools can be improved.

Thank you for your consideration, and I hope this addresses your concern about emphasizing the novelty and importance of the manuscript.

1. The main optimization method that is used for FWI is l-BFGS as it takes 2nd order derivatives into account, but it is not computationally as expensive as BFGS. However, this is not discussed in the Scipy-related section. I encourage the author to focus on this algorithm in the Scipy section and even the other sections.

In our manuscript, Scipy section, as well as the other frameworks sections, primarily focuses on the features of the optimization packages rather than delving into specific optimization algorithms. Including an in-depth discussion of a particular method, such as l-BFGS, might deviate from the central objective of the section, as well the study.

That said, we recognize the relevance of l-BFGS, given its computational efficiency and ability to approximate second-order derivatives without the same computational expense as BFGS. To address your suggestion, we have added a very short description of the l-BFGS method in section 4 of the revised manuscript, highlighting its key features and importance. Furthermore, we recommend readers consult the foundational paper on l-BFGS for a more detailed explanation of how it works and applications.

2. The author should explain what line search methods have been used for each algorithm and each inversion (runs of FWI, Fig. 3.).

Thank you for your suggestion. We have added a paragraph specifying the line search method used by each framework to provide better insight into their convergence behavior and results. This clarification helps contextualize the differences in results/performance observed across the optimization libraries.

3. In section 4, I recommend discussing the reason for choosing l-BFGS, providing examples of other studies using this method or other optimization algorithms.

The primary reason for choosing l-BFGS is its availability across all the optimization frameworks analyzed in our study, and this is already mentioned in the manuscript. While there is no other specific reason for its selection, it is worth noting that l-BFGS is a well-established method, widely recognized for its efficiency in solving large-scale inverse problems, such as FWI.

To further strengthen the manuscript, we have included references to some studies in the literature where l-BFGS has been successfully applied to solve FWI problems.

L31: I don't agree with using the word "quickly" here.

We have revised this part of the text to provide a more precise description without using "quickly."

L149: Modify the sentence, "It also includes includes solvers".

The sentence "It also includes includes solvers" has been corrected to remove the duplicated word "includes."

L314: Please provide some algorithms (as examples) that are available through NLopt.

In response to your suggestion, we have added examples of algorithms available through NLopt, providing further clarity on the options provided by this framework.

L601: The algorithm is called l-BFGS. Please modify the text (lowercase L).

The text has been modified to correctly refer to the algorithm as "l-BFGS," with a lowercase "l," addressing the typo.

Thank you,

Amir

Recommendation: Revisions Required

Round 2

Reviewer A

Dear author,

I appreciate the effort you have put into addressing my comments in the revised manuscript. The extended explanations and modifications to the manuscript have improved the clarity of the study.

After reviewing the updated manuscript, I find that most of my concerns have been addressed. Again, thank you for your efforts.

Best,

Reviewer B

Reviewer Comments

Thank you for inviting me to review this manuscript. The paper presents a comparative review of four open-source full waveform inversion (FWI) toolboxes, briefly introducing their features and evaluating their performance using the Marmousi model. While the study provides a useful preliminary assessment, the analysis lacks depth in several critical aspects that are essential for a comprehensive review.

Key Concerns and Suggestions for Major Revision

1. **Depth of Analysis**
    - The comparison between the four toolboxes remains superficial. A more detailed discussion on their respective strengths, weaknesses, and unique features is necessary.
    - The review should include quantitative benchmarks (e.g., computational efficiency, memory usage) rather than just qualitative descriptions.
2. **Missing Evaluation Criteria**
   The manuscript should expand its assessment to address the following aspects:
    - **Extensibility**: Can the toolboxes be easily modified or extended for new functionalities?
    - **Usability & Installation**: How user-friendly are they? Are there dependency or compatibility issues?
    - **High-Performance Computing (HPC) Compatibility**: Are they easily deployable on supercomputers (e.g., GPU/MPI support)?
    - **Computational & Storage Efficiency**: How do they compare in terms of speed and memory requirements?
    - **Interoperability**: Can they be integrated with other seismic processing or inversion toolboxes?
3. **Broader Implications & Recommendations**
    - The authors should provide clearer guidelines on which toolbox is best suited for different scenarios (e.g., small-scale academic testing vs. large-scale industrial applications).
    - A more systematic benchmarking approach (e.g., standardized tests beyond just the Marmousi model) would strengthen the study.

Recommendation

Given these limitations, I recommend a **major revision** to enhance the depth and rigor of the analysis. The revised version should provide a more thorough, structured, and quantitative comparison to better serve researchers and practitioners in the field.

Response to Reviewers

Review for manuscript

Open-source numerical optimization Python libraries for full-waveform inversion: A Review

By

Oscar Mojica

Round 2

Thank you for sharing the second round of reviews for our manuscript submitted to Seismica. We sincerely appreciate the thoughtful and valuable feedback provided by Reviewer A and Reviewer B.

We observed that Reviewer B's comments in the first round were minor, whereas their second-round feedback requests major revisions with a different scope. This shift led us to consider the possibility that Reviewer B may have changed for this round. To enhance the clarity of the review process, we respectfully suggest that Seismica consider transparently indicating any changes in reviewer assignments in future reviews. Such transparency would help authors better contextualize feedback and respond more effectively.

Additionally, some of Reviewer B's comments exhibit characteristics that may suggest the use of large language model (LLM)-generated text, such as formal phrasing or broad suggestions. We note that Seismica's author guidelines permit the use of AI tools, including LLMs, for enhancing submissions, provided their use is disclosed and content remains the authors' responsibility. While these guidelines apply to authors, we respectfully propose that Seismica consider extending similar clarity to the use of LLMs in the peer review process to ensure consistency across all parties.

Below, we provide a detailed point-by-point response to the reviewers' comments, outlining the changes made in the revised manuscript. The reviewers' specific comments are presented in bold text, followed by our responses in normal text. Thank you for your guidance and for considering our suggestions regarding the review process.

Reviewer A:

Dear author,

I appreciate the effort you have put into addressing my comments in the revised manuscript. The extended explanations and modifications to the manuscript have improved the clarity of the study.

After reviewing the updated manuscript, I find that most of my concerns have been addressed. Again, thank you for your efforts.

Best,

Recommendation: Accept Submission

Thank you for your thoughtful and thorough review of our manuscript and for your recommendation to accept it for publication in Seismica. We are deeply grateful for the time and effort you invested in providing constructive and insightful feedback throughout the review process. Your detailed comments in the first round of revisions were instrumental in identifying areas for improvement, significantly enhancing the clarity, rigor, and overall impact of our work.

Reviewer B:

Thank you for inviting me to review this manuscript. The paper presents a comparative review of four open-source full waveform inversion (FWI) toolboxes, briefly introducing their features and evaluating their performance using the Marmousi model. While the study provides a useful preliminary assessment, the analysis lacks depth in several critical aspects that are essential for a comprehensive review.

Key Concerns and Suggestions for Major Revision

1. **Depth of Analysis**

    - **The comparison between the four toolboxes remains superficial. A more detailed discussion on their respective strengths, weaknesses, and unique features is necessary.**

-We appreciate the reviewer's feedback regarding the depth of our comparison of the four optimization toolboxes. As outlined in the abstract, our analysis was designed to focus on core features, documentation quality, and learning curves, as these are critical factors for practitioners selecting a toolbox for their applications. We believe this approach effectively highlights the strengths and weaknesses of each toolbox implicitly. For instance, the discussion of scipy.optimize's broad algorithm suite, which underscores its versatility (a strength), while noting that its conjugate gradient method is limited to unconstrained problems (a weakness). Similarly, we highlight PyROL's advanced optimization algorithms (a strength) but note its limited Python-specific documentation (a weakness), with comparable analyses for NLopt and sotb-wrapper.

Additionally, to better reflect the focused scope of our comparison, we have revised the manuscript's title to "A Brief Exploration of Open-source Gradient-based Numerical Optimization Python Libraries for Full-waveform Inversion", which emphasizes the exploration of key features of the frameworks. We hope these revisions address the reviewer's expectations. We welcome further guidance if specific aspects require additional emphasis.

- **The review should include quantitative benchmarks (e.g., computational efficiency, memory usage) rather than just qualitative descriptions.**

-To address this, we have incorporated quantitative benchmarks for computational efficiency in the revised manuscript. Specifically, we report that the inversions took an average of 04:01, 05:20, 05:52, and 06:20 minutes when using the scipy.optimize, sotb-wrapper, NLopt, and PyROL frameworks, respectively, on a Dell workstation equipped with an Intel® Xeon® E5-1607 processor (3.00 GHz, four cores, 16 GB RAM) running Ubuntu 22.04, with shots distributed across the four cores.

2. **Missing Evaluation Criteria**
   The manuscript should expand its assessment to address the following aspects:

- **Extensibility: Can the toolboxes be easily modified or extended for new functionalities?**

-This aspect is addressed in the "Discussion and Conclusions" section of our manuscript, where we included a paragraph discussing the evolution and maintenance of these toolboxes. At least three of the toolboxes—scipy.optimize, NLopt, and PyROL—are actively maintained, indicating that the addition of new functionalities is expected as part of their ongoing development. Furthermore, a key advantage of these toolboxes is their open-source nature, which allows users, not only developers or maintainers, to modify or extend the code and distribute it in accordance with their respective licenses. This openness ensures that users with the necessary programming skills can readily adapt or extend the toolboxes to incorporate new features tailored to their specific needs. To clarify this in the manuscript, we have added a brief sentence in the discussion section emphasizing the extensibility of these open-source toolboxes, particularly for users with programming expertise.

- **Usability & Installation: How user-friendly are they? Are there dependency or compatibility issues?**

-The question of how user-friendly these frameworks are is implicitly addressed in our manuscript through the evaluation of documentation quality and the provision of code snippets. Clear and comprehensive documentation, as discussed, directly contributes to user-friendliness by guiding practitioners in effectively utilizing each toolbox. We make a direct mention of that, for example, in the case of sotb.warpper: "*Sotb-wrapper's examples comprehensively encompass the necessary elements for solving optimization problems with the package, making it remarkably user-friendly*."

Additionally, the code snippets provided in the manuscript highlight ease of use, with most toolboxes requiring concise code for solving the optimization problem, reflecting their user-friendly design. Only PyROL requires more lines of code and relies on less explicit Python-specific documentation. Through this evaluation of documentation and code snippets, readers can form a clear idea of which framework is more user-friendly. Furthermore, Python's inherent user-friendly design enhances the accessibility of all evaluated toolboxes. This is noted in the abstract, which states: "*Geoscientists favor Python for its user-friendly interface ...*"

Regarding dependency and compatibility issues, we are not aware of any significant problems, as the toolboxes rely on standard Python libraries (e.g., NumPy) and are easily installed using the pip package manager, which automatically resolves and installs package dependencies. To ensure reproducibility, we provide containerized code (e.g., using Docker), mitigating potential installation challenges.

- **High-Performance Computing (HPC) Compatibility: Are they easily deployable on supercomputers (e.g., GPU/MPI support)?**

-This aspect was addressed in the "Discussion and Conclusions" section of our manuscript, where we discuss the scalability of these frameworks for large-scale problems. As noted, while our study focused on a simplified 2D example, the methodology used can be directly adapted to higher-dimensional problems, such as 3D inversions. Objective function and gradient computation leverage parallel execution by utilizing Dask to efficiently distribute computations across cores on a workstation. Furthermore, we highlighted that frameworks like Dask-Jobqueue can be integrated to scale computations across multiple nodes in an HPC environment, enabling deployment on supercomputers.

- **Computational & Storage Efficiency: How do they compare in terms of speed and memory requirements?**

-In the context of full-waveform inversion (FWI), storage demands are primarily driven by gradient computations, which involve summing contributions over shots and timesteps, while computational efficiency is largely influenced by the line search process, as it may require additional gradient evaluations and forward simulations depending on how it is implemented. Consequently, both storage and computational efficiency are predominantly determined by the characteristics of the FWI problem itself rather than the optimization framework, with the frameworks' contributions to these factors being relatively minor in comparison. In our manuscript, we tested a common optimization method available across all frameworks (e.g., L-BFGS) and detailed its line search implementation to compare computational efficiency. Directly comparing the toolboxes in terms of speed and memory requirements is impractical, as these metrics heavily depend on the specific optimization algorithm and its line-search method employed within each toolbox.

- **Interoperability: Can they be integrated with other seismic processing or inversion toolboxes?**

-There is no inherent impediment to integrating these frameworks with additional packages. They generally require only a function that computes the gradient and objective function or passing the gradient directly. This design allows any seismic processing to be used before or within these functions as needed. Moreover, since much of the seismic processing typically occurs prior to inversion, there are no significant compatibility issues.

3. **Broader Implications & Recommendations**

- **The authors should provide clearer guidelines on which toolbox is best suited for different scenarios (e.g., small-scale academic testing vs. large-scale industrial applications).**

-Any of the evaluated frameworks can be effectively used for both small-scale academic testing and large-scale industrial applications, provided an efficient solver for forward and adjoint computations is available, a critical factor for FWI performance. The choice of framework depends on user preference and project needs, as each toolbox offers distinct capabilities. As demonstrated in the synthetic example, the line-search strategy influences both results and performance, guiding framework suitability based on user priorities.

- **A more systematic benchmarking approach (e.g., standardized tests beyond just the Marmousi model) would strengthen the study.**

-We understand this may refer to incorporating additional synthetic models or real-world field data to broaden the evaluation. As noted in our response to a similar comment in the previous review round, our primary focus in this work is to provide a comparative analysis of optimization libraries using a well-controlled, reproducible setup, with the Marmousi model serving as a standardized and widely recognized benchmark for FWI. Extending the study to include additional synthetic models or field data, while valuable, introduces complexities that are beyond the scope of our current analysis, which aims to maintain a focused comparison of the toolboxes' core functionalities. To facilitate broader benchmarking, we provide our complete experimental setup, including source code and a containerized environment (e.g., using Docker), enabling the community to extend our experiments to other models or datasets. Our FWI implementation uses Devito, which supports code generation for various physical models, including elastic wave propagation, allowing straightforward adaptation for additional scenarios, such as elastic FWI. Devito's official repository includes relevant examples (e.g., https://github.com/devitocodes/devito/blob/master/examples/seismic/tutorials/06_elastic_varying_parameters.ipynb) that can guide such extensions. For real-world field data applications, where additional preprocessing may be required, our codebase, though basic, provides a pathway for implementation.

Recommendation

Given these limitations, I recommend a major revision to enhance the depth and rigor of the analysis. The revised version should provide a more thorough, structured, and quantitative comparison to better serve researchers and practitioners in the field.

Recommendation: Revisions Required

Comments by the Handling Editor

I again thank the author for submitting this manuscript to *Seismica*. The manuscript has been significantly improved, and I also appreciate the revised title, which better reflects the scope of the work. While the authors focus on comparing four optimization frameworks, it would be helpful to briefly mention other Python-based open-source tools for full-waveform inversion, such as PySIT (http://pysit.org/), so that readers are aware these four frameworks are not the only available options.

Regarding the author's question about the peer review process: Reviewer 2 from the first round did not respond to our invitation to review the revised manuscript, so a new reviewer was assigned for the second round. This explains why new suggestions appeared at this stage. It's understandable that different reviewers may approach the manuscript from varying perspectives.

I acknowledge the author's concern about whether the new reviewer may have used a large language model in preparing their review. *Seismica* has policies in place to ensure ethical and respectful peer review. As stated in our reviewer guidelines (https://seismica.library.mcgill.ca/policies#guidelines-for-reviewers):

"Reviewers should ensure that all unpublished data, information, interpretation and discussion in a submitted article (which hasn't been published in a preprint repository) *remain confidential* (See the policies on confidentiality) and may not use reported work in unpublished, submitted articles for their research. Thus, reviewers may not paste text from submitted manuscripts into ChatGPT or similar AI language models when composing reviews."

However, this policy does not prohibit reviewers from using language models to help with grammar or language clarity. What is not allowed is using such tools to interpret or summarize the manuscript content.

Finally, I am pleased to accept this manuscript for publication in *Seismica*. Thank you again for submitting your work.