Response to referees:

HarmonicBalance.jl: A Julia suite for nonlinear dynamics using harmonic balance

First, we would like to thank the referees for their detailed and thorough reviews of our manuscript. We are glad that Referee 1 recognizes the clarity and significance of our work. We also appreciate that Referee 1 finds the results of high practical contribution, and gives valuable, detailed suggestions. The referees' comments allowed us to significantly clarify and improve our manuscript. In the following, we answer (in blue text) to each of the points raised by the referees (in black), and highlight the corresponding changes in the revised manuscript (in red).

Response to Referee 1:

The authors present an open-source software package, suitable for aiding the analysis of many non-linear driven dissipative systems. These systems often have harmonic time-dependent drives, and their steady-states can be found using the method of harmonic balance. This involves making the ansatz that the final solution includes only a finite set of harmonics with slowly varying amplitudes, obtaining a set of slowly varying amplitudes one can study more easily, since the time-scale of the dynamics is now massively increased. Often, one is not interested in the dynamics of these equations, but rather the steady-state configurations and the different operation regimes of the system in question. Doing this requires an analysis of the fixed points of the harmonic equations, which in turn entails the solution of a system of non-linear equations. Here comes the main novelty of tool in question, which builds on another open-source software package - HomotopyContinuation.jl. This allows the authors, in many cases of interest to physics and engineering, to find the entire set of solutions for the harmonic equations in one go. First, I would like to commend the authors for presentation and clarity of the manuscript. The paper is extremely well-written and explains the concepts and their applications in a very concise and clear manner. The figures are clean and informative, and really aide the reading and understanding of the concepts in the paper. It is really a fantastic review of the material, and I am sure many people in the community of driven-systems would benefit from it. As a tool, I find the package to be very useful and convenient. It provides a very useful and versatile analysis tool that would surely be useful to many people in the community.

Functionality wise, the package includes many useful, and very versatile tools. One can scan easily scan any pair of parameters to visualise the configuration state. However, the package still lacks some important tools, especially lacking is the identification of limit-cycles in addition to attractors and unstable point. This is a very central and very common state in many driven systems, and this functionality is severely missing from the package.

We thank the Referee for their appreciation of the functionalities of our package, as well as their comments on additional features. While explicit support for limit cycles has not been presented in the current version of the package, whose aim is the determination and characterisation of isolated fixed points. In principle, limit cycles can be captured in two ways. (i) By seeking time-dependent solutions to the harmonic equations, as opposed to fixed points. This can be achieved in our package by combining the symbolic and time-dependent functionalities, see https://nonlinearoscillations.github.io/HarmonicBalance.jl/dev/examples/time_dependent/. (ii) By expanding the harmonics ansatz to include additional frequencies accounting for the limit cycle motion. This is also feasible here, but raises a question of *how to find these additional frequencies*. We are currently exploring this problem, but since this is at the forefront of current research, we decided to not pursue it in our implementation-focused paper. That said, the package is open source and we will be more than happy for any suggestions, discussions and pull requests regarding this topic.

Action taken:

We revised section 2.2 to stress the envisioned usage of harmonic expansions to find limit cycles:

> Hence, for a system of N interacting components, each expanded in M harmonics, Eq. (8) consists of 2NM harmonic equations. Note that the discussed approach is not exclusive to fixed points, and the ansatz can account for periodic asymptotic orbits, i.e., to limit cycles [1-5, Ref.1, Ref.2, Ref. 3, Ref.4]. Limit cycles can emerge when the system loses stability around Hopf bifurcations and a self-sustained oscillation emerges. We plan to include limit cycle functionality in future releases of the package.

In the revised text, we included four new references:

[Ref. 1]: C. Zambon *et al.* (2020). Parametric instability in coupled nonlinear microcavities, Phys. Rev. A 102, 023526

[Ref. 2]: A. Ganesan, C. Do, and A. Seshia, Phononic Frequency Comb via Intrinsic Three-Wave Mixing, Phys. Rev. Lett. 118, 033903

[Ref. 3]: D. A. Czaplewski *et al.* Bifurcation Generated Mechanical Frequency Comb, Phys. Rev. Lett. 121, 244302

[Ref. 4]: M. I. Dykman, Gianluca Rastelli, M. L. Roukes, and Eva M. Weig, Resonantly Induced Friction and Frequency Combs in Driven Nanomechanical Systems, Phys. Rev. Lett. 122, 254301

In terms of performance, I ran the software package on an M1 MacBook Air. For the Duffing oscillator problem presented in the paper, and for the same parameters and ranges, I've found the performance to be quite close to what's reported in the paper. It would be great, however, to include a performance comparison between HarmonicBalance.jl and other common tools.

We appreciate the Referee's suggestion. We identified and detailed a set of related software packages in section 4 "Comparison with other harmonic balance implementations". Nevertheless, it is difficult to offer an insightful benchmark with these software packages that implement the harmonic balance method, as they are not geared towards finding the complete set of steady states of user-defined equations of motion, which our tool uniquely enables. In particular, it seems that the other packages employ time evolution from a given set of initial conditions or single-root finding methods such as Newton's descent, followed by continuation from a single solution. We instead use homotopy continuation, which turns out to be the computational bottleneck. The performance of HomotopyContinuation.jl has been benchmarked and compared to similar packages, see *Breiding, P., & Timme, S. International Congress on Mathematical Software (pp. 458-465). Springer, Cham.* i.e. Ref.[31].

Action taken:

We have extended the comparison with other harmonic balance implementations, by further elaborating the first bullet point in section 4.

> The use of homotopy continuation allows us to find all possible solutions to the harmonic balance equations, where solutions for multiple parameters can be obtained reliably and efficiently. In this critical step, usually the most computationally intensive, we rely on HomotopyContinuation.jl, a package which outperforms other established homotopy continuation libraries [31].

Some quality of life suggestions -

• *The plotting functions are very useful and convenient, but it would be nice to be able to have more customization options, or even better - have access to the data points directly.*

• A more detailed progress indicator would be very useful, especially for longer calculations. Right now the only progress indicator is the one generated by HomotopyContinuation.jl.

We value the suggestions for improvement of the library, which we implemented in the revised new version of the code. Our plotting routines now allow further customisation options (passing keyword arguments from the plotting library Matplotlib) and data exports (now in the form of a dictionary that contains all necessary data and marker information to replicate the plots). Regarding the second point, we point out that the homotopy continuation algorithm, rather than symbolic manipulation, is almost universally the bottleneck. This is largely due to the first "warm-up" step described in Appendix B, "Solving steady states for multiple parameter values". This step requires tracking many solution paths, often of the order of the maximum number of roots given by Bézout's theorem. In certain cases, the post-processing of solutions (branch matching in particular) may also be quite lengthy.

Action taken:

We have revised our plotting routines as detailed above and upgraded our progress bar to offer a human-readable output during parallel calculations and considering progress along with analysis tasks (matching solution branches, calculating response spectra).

With that said, and as the authors recognize - the tools it uses are published and known, and nothing in the package provides real novelty, even into the Julia framework. I do not want to downplay the importance and utility of the package as a tool, but it does not introduce any new capabilities, not generally, and not even specifically into Julia. None of the tools used in HarmonicBalance.jl are new - the harmonic equations are found straightforwardly using the Symbolic.jl package of Julia, the equations are solved and analyzed using the DifferentialEquations.jl package, and the steady-state configurations are found using HomotopyContinuation.jl package and the graphics are done using the standard graphics libraries of Julia. Generally, HarmonicBalance.jl does not provide any novel computational capabilities or methods that were not known or inaccessible before, rather its main contribution is packaging them all together in a very convenient form, both in terms of the software package and in the very clear explanations and examples given in the manuscript.

To summarize, while the scientific novelty of this manuscript is not high, a fact the authors are very clear and honest about, I believe its practical contribution is high. The software package gives just the right level of specification to make it easy to use, but keeps a level of abstraction to not become hindered when applied to other general problems of interest.

As a scientific paper, it is very well written, but its scientific novelty is quite low. I think it definitely deserves publication, but perhaps not in SciPost Physics. I would recommend publishing it as a review article on SciPost Physics Lecture Notes, or as an acommpanying paper for the package on SciPost Physics Codebases, rather than as a research article.

We thank the Referee for their assessment. Our principal aim in bringing this package to life is to provide an accessible but computationally efficient and robust tool. Numerous fields of study can benefit from a systematic solution for finding the stationary behaviour of nonlinear differential equations. The analysis path/algorithmic workflow that we chose relies on a harmonic balance ansatz. At the same time, we imbue into the code several years of experience in handling the solutions of such systems beyond the initial ansatz. For an efficient implementation of our methodology, we adopted established toolboxes from the Julia ecosystem that were not explicitly interfacing with each other beforehand. For instance, we are not aware of harmonic balance approaches suited for finding the complete steady-state solution landscape of a given nonlinear system. Our harnessing of the homotopy continuation method uniquely enables such a capability.

Our goal with the current submission is to offer a user-friendly tool to address a family of problems rather than claim a specific scientific novelty in physics. Hence, we have decided to follow the Referee's suggestion and have proceeded with a resubmission to SciPost Physics Codebases.