nature portfolio

Peer Review File

An efficient Julia framework for hierarchical equations of motion in open quantum systems

Reviewers' comments:

Reviewer #1 (Remarks to the Author):

As the title indicates, this manuscript accompanies the implementation of the HEOM open quantum systems package in Julia. The manuscript is clearly written and well presented. It includes pedagogical theoretical background to the "hierarchical equations of motion" approach before clearly describing the functionality and capabilities of the software package, and it also comes with two non-trivial practical examples. Further, it includes benchmarking data against a QuTiP HEOM implemention co-developed by some of the authors.

It is clear that the focus of this work is in introducing a novel tool, rather than reporting or analysing original scientific results, although a nice discussion of the results that were obtained is included. The package looks like it could be valuable to the community, a few detailed comments and observations on it follow below:

* This new Julia implementation beats other available HEOM libraries in terms of features; a particular unique highlight is the truncation of environment hierarchy through user-specified importance thresholds.
* It features notable performance enhancements over QuTiP-BoFiN. As the authors acknowledge much of this is due to Julia's inherently superior performance, but this makes it no less valuable to the user.
* This package takes advantage of existing high-performance Julia libraries and integrates well with Julia package ecosystem, and unlocks the benefits of the Julia design philosophy.
* Its user-friendly and intuitive syntax makes constructing and analysing models simple, and the code is both well documented and extensible.
* As enhancements over QuTiP-BoFiN (besides the improved performance), it supports multi-threading for the Liouvillian construction and provides even- and odd-parity Liouvillians for fermionic environments.
* The killer feature may be the easy support for multi-environment situations (both fermionic and simultaneously bosonic) which few other packages and indeed approaches offer so easily.

A couple of minor comments on the manuscript:

* following Eq 5 the introduction of V_s is confusing: it says it can act both on fermionic and bosonic environments, but only appears in Eq 5 which is of the latter type.
* the origin and significance of the even / odd parity propagators could be more clearly and pedagogically explained, perhaps with a simple intuitive example.
* a more detailed discussion of the limits, or limitations, of the approach might be useful. For example, how well does it fare if the system has more than a couple of levels, and could it, for example, deal with two qubits both having an individual bath as well as a shared environment?

In summary, this work introduces an exceptionally well-executed and exciting numerical tool with a list of significant advantages over any of the existing HEOM packages. Of particular note in addition to the high-performance and high-accuracy numerics are the ease of writing and reading code. Overall, this is a welcome addition to the suite of numerical open quantum systems solvers. Being more a "software" than a research publication, it may be for the Editor to decide whether that is in the remit of this journal, but it is clearly an important contribution and a significant achievement and should be published.

Reviewer #2 (Remarks to the Author):

The manuscript by Huang et. al. introduces the Julia software package Heom.jl (HierarchicalEOM.jl) for

solving the hierarchical equations of motion (HEOM) for the reduced dynamics of a system coupled to multiple fermionic and bosonic reservoirs. This is accomplished be defining a hierarchy of auxiliary density operators (ADOs) that model interactions and resulting entanglement between a system and the corresponding coupled baths. Significantly, the HierarchicalEOM.jl introduces an importance value associated with each ADO, and an importance value cutoff, with all ADOs with an importance value less than the cutoff being discarded. Compared to similar packages, this functionality is unique to HierarchicalEOM.jl, and provides a useful tool for controlling computational complexity with a well-controlled approximation. This manuscript clearly demonstrates the utility of the HierarchicalEOM.jl software package. Therefore, I recommend this manuscript for publication. My report continues below, containing additional discussion, as well as a few suggestions for authors to help improve the usability of the package.

By implementing the HierarchicalEOM.jl package in the Julia programming language, they introduce a tool that has a convenient user interface while remaining computationally efficient, as demonstrated by benchmark results comparing their package to QuTiP-BoFiN. As a registered Julia package, HierarchicalEOM.jl is easy to install using the built-in Julia package manager. Moreover, their software development workflow uses best practices, leveraging continuous integration (CI) to automate both unit testing and online documentation generation.

In the manuscript they introduce as an example use-case a model for the Kondo effect whereby a single-level electronic system is coupled to a pair of fermionic reservoirs (left and right), as well as a single-mode cavity, which itself is then coupled to a bosonic reservoir. Using the HierarchicalEOM.jl package, they calculate the electronic density of states, conductance and electronic current as a function of applied bias voltage, as well as the power spectral density for the single-mode cavity. This is a non-trivial example that demonstrates the utility of their package.

Below I list some minor suggestions for how the manuscript and software package might be improved:

- An implementation of the examples discussed in the manuscript is linked via a separate github repository Heom.jl-Examples. However, it would be convenient if these examples were integrated directly into the online documentation. That said, it would be a bad idea to directly include Jupyter notebooks in the HierarchicalEOM.jl github repository as this can quickly lead to bloat in terms of the repository memory footprint. A viable workaround is to include an examples tab in their online documentation, and to then use the package Literate.jl (https://github.com/fredrikekre/Literate.jl.git) to manage and maintain the examples. This package allows users to write annotated Julia scripts that can then be exported as either a notebook, markdown file, or plain Julia script. The export process can be integrated into the CI process. Moreover, this allows any example included in the documentation to also double as integration test in the unit testing. To see an example of a package using Literate.jl to export examples in documentation, I refer the authors to the Sunny.jl package (https://github.com/SunnySuite/Sunny.jl.git).

- My next suggestion is to simply include more examples in the online documentation. Specifically, including examples demonstrating how to integrate HierarchicalEOM.jl with existing Julia packages, like DifferentialEquations.jl and LinearSolver.jl, would be especially nice.

- It appears you did your plotting using the Plots.jl package. I would recommend you consider using the Makie.jl (https://docs.makie.org/stable/) package instead of, or in addition to, the Plots.jl package. In my view, the Makie.jl package has better documentation, more extensive examples, and is being more actively being developed and maintained.

- Lastly, I found it somewhat confusing that you refer to the package as HEOM.jl in the manuscript, even though the package is registered under the name HierarchicalEOM.jl. I would recommend simply referring to the package as HierarchicalEOM.jl throughout the manuscript. I cannot help but notice that this is what you did in your arXiv post (https://arxiv.org/abs/2306.07522).

Response of the First Reviewer

Reviewer's Comment

As the title indicates, this manuscript accompanies the implementation of the HEOM open quantum systems package in Julia. The manuscript is clearly written and well presented. It includes pedagogical theoretical background to the "hierarchical equations of motion" approach before clearly describing the functionality and capabilities of the software package, and it also comes with two non-trivial practical examples. Further, it includes benchmarking data against a QuTiP HEOM implemention co-developed by some of the authors.

It is clear that the focus of this work is in introducing a novel tool, rather than reporting or analysing original scientific results, although a nice discussion of the results that were obtained is included. The package looks like it could be valuable to the community, a few detailed comments and observations on it follow below:

- *This new Julia implementation beats other available HEOM libraries in terms of features; a particular unique highlight is the truncation of environment hierarchy through user-specified importance thresholds.*
- *It features notable performance enhancements over QuTiP-BoFiN. As the authors acknowledge much of this is due to Julia's inherently superior performance, but this makes it no less valuable to the user.*
- *This package takes advantage of existing high-performance Julia libraries and integrates well with Julia package ecosystem, and unlocks the benefits of the Julia design philosophy.*
- *Its user-friendly and intuitive syntax makes constructing and analysing models simple, and the code is both well documented and extensible.*
- *As enhancements over QuTiP-BoFiN (besides the improved performance), it supports multi-threading for the Liouvillian construction and provides even- and odd-parity Liouvillians for fermionic environments.*
- *The killer feature may be the easy support for multi-environment situations (both fermionic and simultaneously bosonic) which few other packages and indeed approaches offer so easily.*

A couple of minor comments on the manuscript:

Our Response

We thank the Reviewer for the careful reading of our manuscript and constructive comments. In the following, we provide a detailed response and corresponding updates to the manuscript.

Reviewer's Comment 1.

following Eq. 5 the introduction of V_s is confusing: it says it can act both on fermionic and bosonic environments, but only appears in Eq. 5 which is of the latter type.

Our Response

We thank the Reviewer for pointing out the potential ambiguity related to the interaction operator V_s. We understand the confusion and apologize for any lack of clarity in our initial presentation.

In Eq. (5), we focus on the interaction between an arbitrary system and bosonic environment. Here the system can be anything; bosonic, fermionic, or spin-like. In particular, as discussed in other works, such as arXiv:2302.01044 (2023), Phys. Rev. B 98, 075105 (2018), and J. Chem. Phys. 138, 214111 (2013), V_s, the interaction operator, can represent a fermionic system, but must conserve fermionic excitation number when coupled to a bosonic environment. Therefore, in a more general context, V_s can act on either fermionic, bosonic or spin degrees of freedom for the system, or even all three.

Corresponding Manuscript Changes

To make this point clearer, we have expanded the explanation about the role and generality of V_s in Eq. (5) in the manuscript. We have also revised Figure 1.

Reviewer's Comment 2.

the origin and significance of the even / odd parity propagators could be more clearly and pedagogically explained, perhaps with a simple intuitive example.

Our Response

We appreciate the suggestion to improve our explanation of the even and odd parity propagators. In the manuscript, we introduce the propagators for even ($p = +$) and odd ($p = -$) parity operators to keep track of the parity of the input operators throughout the time evolution. The symbols $[\cdot,\cdot]_-$ and $[\cdot,\cdot]_+$ denote the commutator and anti-commutator, respectively. Furthermore, $-p = -(\pm) = \mp$ stands for the projection onto the odd ($-$) or even ($+$) parity sector.

The dependence of the propagator on the parity of the underlying state is an interesting consequence of the structure of the Hilbert space in composite fermionic systems. In fact, since fermionic operators acting on independent parts of the Hilbert space do not necessarily commute, special care is required in the way the environment is traced out, i.e., in the definition of the reduced density matrix. As an example, for an environment made out of a single fermion, the reduced matrix elements $\langle i|\rho_{\rm s}^{p}|j\rangle$ (in a basis labeled by $\langle i|$ and $|j\rangle$) involve the perturbative sum of expressions of the form $\langle i|(c\tilde{\rho}_{\rm e}\tilde{\rho}_{\rm s}^{p}c^{\dagger} + \tilde{\rho}_{\rm e}\tilde{\rho}_{\rm s}^{p})|j\rangle$ (in terms of environmental operators $\tilde{\rho}_{\rm e}$, system operators $\tilde{\rho}_{\rm s}^{p}$ with parity p, and the environment-annihilation operator c). These quantities depend on the commutator between $\tilde{\rho}_{\rm s}^{p}$ and c, which is trivial only for even parity ($p = +$). However, when the state of the system has odd-parity ($p = -$), the partial trace over the environment requires extra anti-commutations to be accounted for [see, for example, Phys. Rev. B 94, 155142 (2016) or Phys. Rev. B 105, 035121 (2022)]. In the open quantum system setting analyzed here, this results in extra minus signs in the expression for the effective propagator describing the reduced dynamics [as described, more formally, in Phys. Rev. B 94, 155142 (2016)].

It is important to note that, here, by parity we do not refer to the presence of an odd or even number of fermions in the system but, rather, to the number of fermionic operators needed to represent the density matrix itself.

In this sense, a physical density matrix $\rho_{\rm s}^{+}$ always belongs to the even parity sector and the mentioned extra minus-signs only appear in the analysis of the dynamics of operators in the form $\rho_{\rm s}^{-} = d_{\rm s}\rho_{\rm s}^{+}$. While unphysical, these odd-parity quantities are relevant when computing correlations in which system operators are considered at different times, see for example, the Methods section: Numerical Computation of the Spectrum.

Corresponding Manuscript Changes

We included an additional explanation about the even and odd parity propagators after Eq. (7-9) in our revised version of the manuscript.

Reviewer's Comment 3.

a more detailed discussion of the limits, or limitations, of the approach might be useful. For example, how well does it fare if the system has more than a couple of levels, and could it, for example, deal with two qubits both having an individual bath as well as a shared environment?

Our Response

We thank the Reviewer for raising this important question about the limitations of the hierarchical equations of motion (HEOM) approach. To derive the HEOM in the form used in this work, the following three assumptions are necessary.

1. The system and the environments must be initialized in a separable state.

2. Each bath must be initially at thermal equilibrium and described by either a Fermi-Dirac or Bose-Einstein distribution, depending on its statistics.

3. The bath operator within the system-bath interaction Hamiltonian should be linear in the bath annihilation and creation operators.

For example, we do not allow the initial presence of system-environment correlations or baths initially not in thermal equilibrium thereby constituting an intrinsic limitation of our derivation of the HEOM.

One significant advantage of the HEOM method is its versatility; it is applicable to a diverse range of systems and allows interactions to multiple baths, which can be either bosonic or fermionic. It also allows to model complex setups such as composite systems made out of several energy levels and possibly sharing common environments. The computational cost of the simulation increases with the tier of the HEOM hierarchy which depends on the strength of the system-bath interaction and on the system Hamiltonian.

For instance, in "Example 1", the system is made out of just one impurity (consisting of a single spin). Despite its simplicity, this system can allow a strong non-linear repulsion energy U thereby requiring a higher truncation tier. This is essential to capture the correct hybridization between the system and the bath even in the presence of non-linear interactions within the system. Similarly, additional computational costs would also be required in the presence of strong system-bath interactions, as seen in Example 1 within the Kondo regime.

In "Example 2," we model an even more complex situation in which an effective two-level system connected to two fermionic baths is also strongly coupled to a single-mode cavity which interacts with its own bosonic bath. The simulation of this system is challenging, especially when light-matter are ultra-strongly coupled and within the Kondo regime arXiv:2302.01044.

A package like *HierarchicalEOM.jl* in Julia was written with the intention to address these computational challenges, which allow researchers to explore complex physical open quantum systems. To allow a versatile use of this package in the widest possible range of audience, we enriched our online documentation with four pedagogical examples.

Users can now access these examples from the newly-added "Examples" section in the online documentation sidebar at https://ncku-qfort.github.io/HierarchicalEOM.jl/stable/. The topics covered by these examples include:

- Cavity quantum electrodynamics (QED) systems,

- Driven systems and dynamical decoupling,

- The single-impurity Anderson model,

- Electronic current.

Together with the two examples already present in the "Quick Start" section, we believe these six comprehensive examples will enhance the user-friendliness and utility of our *HierarchicalEOM.jl* package.

To conclude, our package offers an implementation of the HEOM method to model complex physical systems while balancing computational costs and accuracy within a user-friendly interface.

Corresponding Manuscript Changes

We outlined the assumptions implicit in our derivation of the HEOM in the manuscript.

Response of the Second Reviewer

Reviewer's Comment

The manuscript by Huang et. al. introduces the Julia software package Heom.jl (HierarchicalEOM.jl) for solving the hierarchical equations of motion (HEOM) for the reduced dynamics of a system coupled to multiple fermionic and bosonic reservoirs. This is accomplished be defining a hierarchy of auxiliary density operators (ADOs) that model interactions and resulting entanglement between a system and the corresponding coupled baths. Significantly, the HierarchicalEOM.jl introduces an importance value associated with each ADO, and an importance value cutoff, with all ADOs with an importance value less than the cutoff being discarded. Compared to similar packages, this functionality is unique to HierarchicalEOM.jl, and provides a useful tool for controlling computational complexity with a well-controlled approximation. This manuscript clearly demonstrates the utility of the HierarchicalEOM.jl software package. Therefore, I recommend this manuscript for publication. My report continues below, containing additional discussion, as well as a few suggestions for authors to help improve the usability of the package.

By implementing the HierarchicalEOM.jl package in the Julia programming language, they introduce a tool that has a convenient user interface while remaining computationally efficient, as demonstrated by benchmark results comparing their package to QuTiP-BoFiN. As a registered Julia package, HierarchicalEOM.jl is easy to install using the built-in Julia package manager. Moreover, their software development workflow uses best practices, leveraging continuous integration (CI) to automate both unit testing and online documentation generation.

In the manuscript they introduce as an example use-case a model for the Kondo effect whereby a single-level electronic system is coupled to a pair of fermionic reservoirs (left and right), as well as a single-mode cavity, which itself is then coupled to a bosonic reservoir. Using the HierarchicalEOM.jl package, they calculate the electronic density of states, conductance and electronic current as a function of applied bias voltage, as well as the power spectral density for the single-mode cavity. This is a non-trivial example that demonstrates the utility of their package.

Below I list some minor suggestions for how the manuscript and software package might be improved:

Our Response

We thank the Reviewer for carefully reading our manuscript and the constructive comments. In the following, we provide a detailed response and outline the corresponding updates to the manuscript.

Reviewer's Comment 1.

An implementation of the examples discussed in the manuscript is linked via a separate github repository Heom.jl-Examples. However, it would be convenient if these examples were integrated directly into the online documentation. That said, it would be a bad idea to directly include Jupyter notebooks in the HierarchicalEOM.jl github repository as this can quickly lead to bloat in terms of the repository memory footprint. A viable workaround is to include an examples tab in their online documentation, and to then use the package Literate.jl (https://github.com/fredrikekre/Literate.jl.git) to manage and maintain the examples. This package allows users to write annotated Julia scripts that can then be exported as either a notebook, markdown file, or plain Julia script. The export process can be integrated into the CI process. Moreover, this allows any example included in the documentation to also double as integration test in the unit testing. To see an example of a package using Literate.jl to export examples in documentation, I refer the authors to the Sunny.jl package (https://github.com/SunnySuite/Sunny.jl.git).

Our Response

We thank the Reviewer for the valuable advice and for providing us with an explicit example package *Sunny.jl* for the use of *Literate.jl*.

We have now expanded our online documentation with four additional pedagogical examples. These examples are all written in annotated Julia scripts and exported using *Literate.jl* as suggested by the Reviewer. We have also integrated the export routine into the GitHub CI process. These examples are now further available through the new section (Examples) in the sidebar of the online documentation (https://ncku-qfort.github.io/HierarchicalEOM.jl/stable/),

where they are listed as follows:

- Cavity quantum electrodynamics (QED) systems,
- Driven systems and dynamical decoupling,
- The single-impurity Anderson model,
- Electronic current.

We hope these six examples (where we included the two examples in the "Quick Start" section) will make our package *HierarchicalEOM.jl* more accessible to the users.

Regarding the source codes for the two examples presented in the manuscript, we changed the file type from "jupyter notebook" to "annotated Julia script". However, we would like to point out that these examples do require substantial computational resources (large memory usage and computational time even for a single data point), and thus, their execution on GitHub CI might not always be feasible.

Furthermore, according to the *Data availability statements and data citations policy* from *Communications Physics*, we believe it would be better to separate the source codes of the examples (in the manuscript) from the new pedagogical examples (in the online documentation), and put them in an individual GitHub repository. This will facilitate the reproduction of our results using the source code in the "Data Availability" section which should no longer be updated.

Reviewer's Comment 2.

My next suggestion is to simply include more examples in the online documentation. Specifically, including examples demonstrating how to integrate HierarchicalEOM.jl with existing Julia packages, like DifferentialEquations.jl and LinearSolver.jl, would be especially nice.

Our Response

We thank the Reviewer for the valuable advice. As we mentioned in the previous response, we have added four extra examples in the online documentation. Based on some of these examples, we have further expanded the documentation by including two benchmark tests for both *DifferentialEquations.jl* solvers and *LinearSolve.jl* solvers. These tests are now available in a new section (Benchmark Solvers) now included in the sidebar of the online documentation (https://ncku-qfort.github.io/HierarchicalEOM.jl/stable/). The benchmark not only demonstrates how *HierarchicalEOM.jl* is integrated with *DifferentialEquations.jl* and *LinearSolve.jl*, but also reveals the differences between each solvers.

Reviewer's Comment 3.

It appears you did your plotting using the Plots.jl package. I would recommend you consider using the Makie.jl (https://docs.makie.org/stable/) package instead of, or in addition to, the Plots.jl package. In my view, the Makie.jl package has better documentation, more extensive examples, and is being more actively being developed and maintained.

Our Response

We express our gratitude to the Reviewer for their insightful recommendation on the use of the *Makie.jl* package. We acknowledge the advantages of *Makie.jl*, especially in terms of its comprehensive documentation, extensive number of examples, and its active development trajectory.

In light of this, we commit to familiarizing ourselves with *Makie.jl* and integrating it into our future works. This approach will not only enhance our data visualization capabilities but also align our research with the continuously evolving best-practices in the community.

We genuinely appreciate the constructive feedback and will take steps to implement it in our forthcoming publications.

Reviewer's Comment 4.

Lastly, I found it somewhat confusing that you refer to the package as HEOM.jl in the manuscript, even though the package is registered under the name HierarchicalEOM.jl. I would recommend simply referring to the package as HierarchicalEOM.jl throughout the manuscript. I cannot help but notice that this is what you did in your arXiv post (https://arxiv.org/abs/2306.07522).

Our Response

We thank the Reviewer for indicating the ambiguity of the package name as we submitted our manuscript before registering the package to Julia. Since our original package name (HEOM.jl) does not meet the registration guidelines for a Julia package, we changed our package name to *HierarchicalEOM.jl*.

Corresponding Manuscript Changes

Throughout the maniscript, we changed our package name from *HEOM.jl* to *HierarchicalEOM.jl*. s

Summarized changes in the manuscript

1. Throughout the manuscript, we changed our package name from "HEOM.jl" to "HierarchicalEOM.jl".
2. We provided an expanded explanation about the role and generality of the coupling operator V_{s} in Eq. (5).
3. We revised Figure 1.
4. We included an additional explanation about the even and odd parity propagators after Eq. (7-9).
5. We outlined the assumptions implicit in our derivation of the HEOM method in the main text.
6. We updated the hyper-links provided in "Data Availability" and "Code Availability" sections.

REVIEWERS' COMMENTS:

Reviewer #1 (Remarks to the Author):

Having carefully gone over the authors' reply and revised manuscript I am very happy to recommend publication of this version: I would like to thank the authors for their detailed and constructive replies which fully addressed all of my questions (as well as those of the other review as far as I can tell).

Reviewer #2 (Remarks to the Author):

The authors have adequately addressed my questions and suggestions for the manuscript. I believe the documentation for HierarchicalEOM.jl is now significantly improved, especially with the addition of more extensive examples, making using the software much simpler for new users. I know this is the case for myself personally, as downloading and using their package was much simpler with the introduction of these new examples. Also, any suggestions I made that were not adopted were well justified. Finally, the manuscript itself has also been significantly improved with their latest revisions. Therefore, I believe this paper warrants publication as is.