**Authors**

**Leading authors:**

Xiao Zheng, Rui-Xue Xv, YiJing Yan.

**Authors (in the alphabetical order):**

Daochi Zhang, Dong Hou, Houdao Zhang, Jiaan Cao, Jingshuang Jin, Lei Cui, Lu Han, Lyuzhou Ye, Zihao Chen.

**Citing This Project:**

**If you find this project useful, then please cite:**

Ye L., Wang X., Hou D., Xu R.-X., Zheng X., Yan Y. HEOM-QUICK: a program for accurate, efficient, and universal characterization of strongly correlated quantum impurity systems.

WIREs Computational Molecular Science. 2016;6(6):608-38.

**and**

Daochi Zhang, Lyuzhou Ye, Jiaan Cao, Yao Wang, Rui-Xue Xu, Xiao Zheng, YiJing Yan. HEOM-QUICK2: a general-purpose simulator for fermionic many-body open quantum systems -- An Update. arXiv2401.01715.

which may also be downloaded from: <https://arxiv.org/abs/2401.01715>.

**ABOUT HEOM-QUICK2:**

Accurate characterization of correlated electronic states, as well as their evolution under external fields or in dissipative environment, is essentially important for understanding the properties of strongly correlated transition-metal materials involving spin-unpaired d or f electrons. This paper reviews the development and applications of a numerical simulation program, the Hierarchical Equations of Motion for QUantum Impurity with a Correlated Kernel (HEOM-QUICK), which allows for an accurate and universal characterization of strongly correlated quantum impurity systems. The HEOM-QUICK program implements the formally exact HEOM formalism for fermionic open systems. Its simulation results capture the combined effects of system-environment dissipation, manybody interactions, and non-Markovian memory in a nonperturbative manner. The HEOM-QUICK program has been employed to explore a wide range of static and dynamic properties of various types of quantum impurity systems, including charge or spin qubits, quantum dots, molecular junctions, and so on. It has also been utilized in conjunction with first-principles methods such as density-functional theory methods to study the correlated electronic structure of adsorbed magnetic molecules. The advantages in its accuracy, efficiency, and universality have made the HEOM-QUICK program a reliable and versatile tool for theoretical investigations on strong electron correlation effects in complex materials.

Since the release of HEOM-QUICK, our focus has been on improving the efficiency and accuracy of the HEOM method. We have developed advanced HEOM methods and the corresponding numerical algorithms that are specifically designed to accurately and comprehensively characterize the many-body correlation effects and non-Markovian memory. The integration of these advancements with the previous program gives rise to the latest fermionic HEOM simulator, HEOM-QUICK version 2 (HEOM-QUICK2).

HEOM-QUICK2 is a Fortran-based open-source program for simulating the dynamics of open quantum systems and is freely available for use and/or modification on the Linux platform. Most codes follow the Fortran95 standard, expect the codes of iteration algorithms in the F77 format. The library of HEOM-QUICK2 depends on the excellent BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage). The program also supports multi-threading with the Open Multi-Processing (OpenMP) framework to boost matrix calculations on shared memory computers.

HEOM-QUICK2 follows a procedural programming paradigm, decomposing the workflow of numerically solving HEOM into three modules: the input-output module, preparation module, and calculation module. Each module consists of various subroutines, which are sequentially called in the main program. HEOM-QUICK2 first reads input information (e.g. system and environmental parameters, external fields, job control tags, etc.) through the input/output (IO) module and feeds them to the preparation module where these input parameters are processed and necessary information about the setting of simulation is generated for subsequent computations. The computation module implements accurate and efficient algorithms to solve stationary states and dissipative dynamics for the given system. The program also employs the obtained reduces density operator (RDO) and auxiliary density operators (ADOs) to calculate local observables and response properties. Finally, the IO module generates auxiliary files which save the details of the above workflow and calculated results.

HEOM-QUICK2 has the following important features:

1. Inherits the strengths of HEOM-QUICK: Enable exploring a variety of many-body OQSs in the existence of diversified external fields with arbitrary time dependence (such as magnetic field, gate voltage, bias voltage, and temperature gradient); Evaluate a variety of local observables and response properties for both equilibrium and nonequilibrium scenarios; Support the user-defined system models and external fields.
2. Efficiently and precisely unravels the pronounced non-Markovian memory: New Fano and Prony spectrum decomposition schemes demonstrate superior numerical performance in low-temperature environments, compared with the Padé spectrum decomposition scheme utilized in HEOM-QUICK. This advancement allows for the exploration of many-body OQSs coupled to much lower-temperature environments.
3. Significantly reduces the computational time and costs for solving stationary states and enhances numerical stabilities for long-time dynamics simulations of strongly correlated many-body OQSs.
4. Extends the applicability of fermionic HEOM method: Enable calculating the time-dependent response properties of many-body OQSs; Significantly enhances the “energy resolution” for low-energy excitations to the sub-meV ( meV) level; Combined with quantum chemistry software, HEOM-QUICK2 can precisely reproduce low-energy spin-flip excitation signatures and spin relaxation dynamics for realistic single atom/molecule junctions experimentally measured in the SP-STM setup; Explores quantum thermodynamics and thermoelectric transport in model systems.

**Requirement**

**For the compilation of HEOM-QUICK2 one needs:**

* Compilers for Fortran (at least F2008 compliant). Recommend [intel-oneapi-base-kit](https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#base-kit)+[intel-oneapi-hpc-kit](https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#hpc-kit);
* Numerical libraries: BLAS, and LAPACK. Recommend [intel-oneapi-mkl](https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#base-kit);
* Support for OpenMP libraries (at least OpenMP4.0) that provide parallel programming framework for Fortran. Recommend [intel-Fortran-compiler-classic](https://www.intel.com/content/www/us/en/developer/tools/oneapi/fortran-compiler.html).

**Installation**

**Step 1: Download**

Download the source code of HEOM-QUICK2, copy it to the desired location on your machine, unzip the file to obtain the folder “/path/to/HEOM-QUICK2.x.x.x” and reveal its content.

**Step 2: Prepare Makefile**

Open the Makefile in “/path/to/HEOM-QUICK2.x.x.x” and modify the required information:

**a)** “**NAME=**/path/to/HEOM-QUICK2.x.x.x/bin/HEOM-QUICK2.x” corresponds to the location where the executable “HEOM-QUICK2.x” is generated;

**b)** “**F77**” defines the command to invoke your Fortran compiler (e.g. gfortran, ifort, …) and “**FFLAGS**” specifies the compile flags. For example, the tag “-qopenmp” tells the parallelizer to generate a multi-threaded executable based on OpenMP directives in the Linux platform;

**c)** “**LIBDIR**” and “**LIBS**” provides the links to BLAS, LAPACK libraries that are a part of intel Math Kernel Library (MKL).

**Step 3: Make**

Build HEOM-QUICK2 with “make all”. The executable is generated in the location described by “NAME”.

**Step 4: Install**

Copy the executable to the system “$PATH” or append “/path/to/HEOM-QUICK2.x.x.x/bin/” defined by “NAME” to the environment variable with the command “export PATH=$PATH:/path/to/HEOM\_QUICK2.x.x.x/bin/” in your “~/.bashrc”.

**Input&Output files**

As a minimal setup, HEOM-QUICK2 only requires a single main input file, i.e. a user-named input file which includes the information of system Hamiltonian, statistic properties of environments and job control tags. This main input file is a tagged format ASCII file. HEOM-QUICK2 calculations are often continued on top of a previous HEOM-QUICK2 calculation. So, in case a calculation is restated, the output files of the previous calculation can be input files for the next calculation. For instance, the “rho\_spa(\_td).sav” file which can serve as an initial state in the sequent calculation, the “coefindex.data” file which provides the coefficients in up- and down-tier operations, the “TAPE\_(tt).resume” file which saves RDO and ADOs in time propagation when time is equal to “tt”, etc. Finally, there is a special input file to induce an instant beak of the calculation: the stoptd(st) file. It is not used in a standard workflow, but it might be convenient to stop a time propagation (steady-state calculation) manually when it takes too long or a technical issue on the compute engine arises.

The main output file of HEOM-QUICK2 is a user-named output file. Here is a comprehensive list of some important output files:

|  |  |  |
| --- | --- | --- |
| **Filename** | **Format** | **Purpose** |
| user-named input file | ASCII | main input file |
| user-named output file | ASCII | general output information |
| indextable.tmp | Binary | scratch file for indextable of RDO&ADOs |
| curr.data | ASCII | time vs. electric current |
| indextable.sav | Binary | backup indextable |
| popu.data | ASCII | time vs. impurity population |
| coefindex.tmp | Binary | scratch file for building indextable to save coefficients in up- and down-tier operations |
| coefindex.data | Binary | backup coefindex to read, readable in next calculation |
| corr.data | ASCII | time vs. bath correlation functions  (used by residue correction) |
| TAPE\_(tt).resume | Binary | time evolution job outputs the values of RDO&ADOs when time is equal to “tt”, readable in next calculation |
| rhodiag.data | ASCII | time vs. diagonal elements of real part of RDO |
| rhotime.data | ASCII | time vs. upper triangular elements of RDO |
| stoptd(st) | ASCII | flag file to stop time propagation (steady-state calculation) |
| poccdetail.data | ASCII | time vs. spin-specific impurity population |
| auxindex.data | Binary | auxiliary index file, readable in next calculation |
| rho\_spa(\_td).sav | Binary | the final result of RDO&ADOs in sparse format, readable in next calculation |
| rho\_spa.gr(.st) | Binary | steady state calculation outputs the final result of RDO&ADOs in sparse format, readable in next calculation |
| sparse\_index.data | Binary | number of nonzeros and index of nonzeros for steady-state calculation |
| sparse\_info.data | Binary | (row,col) position of nonzeros for steady-state calculation |
| sparse\_index\_cf.data | Binary | number of nonzeros and index of nonzeros for solving system correlation function |
| sparse\_info\_cf.data | Binary | (row,col) position of nonzeros for solving system correlation function |
| sparse\_index\_td.data | Binary | number of nonzeros and index of nonzeros for time evolution |
| sparse\_info\_td.data | Binary | (row,col) position of nonzeros for time evolution |
| sdot1.data | ASCII | time vs. x,y,z-spin moment of impruity 1 |
| sdot2.data | ASCII | time vs. x,y,z-spin moment of impruity 2 for multi-impurity system |
| spin\_ddot.data | ASCII | time vs. spin properties of multi-impurities |
| rho\_spa.jac | Binary | final results of RDO&ADOs given by Jacobi iteration in sparse format, readable in next calculation |
| rho\_spa.chk | Binary | intermediate values of RDO&ADOs given by TFQMR iteration in sparse format, readable in next calculation |
| engy.data | ASCII | time vs. internal energy, hybridization energy and the sum of these two |
| ams\_fermion.data | ASCII | system annihilation operators |
| ham\_sys.data | ASCII | system Hamiltonian |
| res\_corr.data | ASCII | results of spectral decomposition |
| RDO\_and\_ADO.data | ASCII | RDO&1st-tier ADO in sparse format |

HEOM-QUICK2 offers Python programs in “/path/to/HEOM-QUICK2.x.x.x/tools/read\_

output\_para” to read “ams\_fermion.data”, “ham\_sys.data” and “res\_corr.data”.

**How to run HEOM-QUICK2**

For beginners we recommend to run the following useful examples to learn the basic operations on HEOM-QUICK2:

**Ex1. solve steady state of single-impurity Anderson model (SIAM) subjected to bias voltage**

We first show an example which employs the TFQMR iterative approach to solve the steady state of SIAM connected to two reservoirs subjected to constant bias voltages. In this example, we unravel the bath correlation functions by the Prony fitting spectrum decomposition scheme and truncate the hierarchy by the adiabatic scheme.

The input file is

1 2

2 4

3 0

4 1

5 2

6 2

7 5.0d0 5.0d0

8 0.4d0 0.4d0

9 0.001d0 0.001d0

10 0.001d0 0.001d0 -0.001d0 -0.001d0

11 1.d3

12 1.d-2

13 $para1 eup=-1.0d0 edown=-1.0d0 uu=2.0d0 $end

14 $field fieldtype=0 $end

15 1.d-20 1.d-20 1.d-20 1.d-20

16 $jobinfo lsparse=T psfjob=T itype\_psf=1 $end

17 $converge maxit0=20000 crit=1.d-7 $end

18 $method methodss=2 $end

19 $adiabatic lad=T $end

It is noted that the standard input file does not require the line numbers in the left side.

Users can run HEOM-QUICK2 with the command “./path/to/HEOM-QUICK2.x.x.x/bin/HEOM-QUICK2.x <input\_file\_name> out\_file\_name”.

**Ex2. calculate linear response properties of a system in steady state**

Sincewe have obtained the steady state in above example, we will next calculate linear response properties of the SIAM, including system correlation function, Green’s function, self-energy due to electron-electron interactions and impurity spectral function in frequency domain. The bath correlation functions by the Prony fitting spectrum decomposition scheme and truncate the hierarchy by the adiabatic scheme.

The input file is

1 3

2 4

3 0

4 1

5 2

6 2

7 5.0d0 5.0d0

8 0.4d0 0.4d0

9 0.001d0 0.001d0

10 0.001d0 0.001d0 -0.001d0 -0.001d0

11 1.d3

12 1.d-2

13 $para1 eup=-1.0d0 edown=-1.0d0 uu=2.0d0 $end

14 $field fieldtype=0 $end

15 1.d-20 1.d-20 1.d-20 1.d-20

16 $jobinfo lsparse=T psfjob=T itype\_psf=1 $end

17 $converge maxit0=20000 crit=1.d-7 $end

18 $method methodss=2 $end

19 $adiabatic lad=T $end

20 $dos ldos=T iorbs\_dos=1 ispin\_dos=1 lfreq\_dos=T

21 freq\_dos= xxxx maxit\_dos=20000 crit\_dos=1.d-7 $end

It is noted that the standard input file does not require the line numbers in the left side.

Users can run HEOM-QUICK2 with the command “./path/to/HEOM-QUICK2.x.x.x/bin/HEOM-QUICK2.x <input\_file\_name> out\_file\_name”.

**Ex3. time propagation for SIAM subjected to *ac* voltage**

Now we impose a sinusoidal *ac* voltage on the SIAM in a steady state calculated in Ex1 and simulate the resulting dissipation dynamics of SIAM. The bath correlation functions by the Prony fitting spectrum decomposition scheme and truncate the hierarchy by the adiabatic scheme.

The input file is

1 1

2 4

3 0

4 1

5 2

6 2

7 5.0d0 5.0d0

8 0.4d0 0.4d0

9 0.001d0 0.001d0

10 0.001d0 0.001d0 -0.001d0 -0.001d0

11 2.5d2

12 5.d-3

13 $para1 eup=-1.0d0 edown=-1.0d0 uu=2.0d0 $end

14 $field fieldtype= 1 lreadomega = T $end

15 0.06d0 0.06d0 0.06d0 0.06d0

16 $jobinfo lsparse=T psfjob=T itype\_psf=1 $end

19 $adiabatic lad=T $end

20 $resume icont=0 lresume=T nresume=2000 $end

It is noted that the standard input file does not require the line numbers in the left side.

Users can run HEOM-QUICK2 with the command “./path/to/HEOM-QUICK2.x.x.x/bin/HEOM-QUICK2.x <input\_file\_name> out\_file\_name”.

**Ex4. calculate time-dependent linear response properties of a system**

HEOM-QUICK2 generates a series of “TAPE\_(tt).resume” files which records the intermediate results of RDO&ADOs during time propagation in Ex3. To evaluate time-dependent linear response properties of the system, we should rename one of “TAPE\_(tt).resume” as “TAPE.resume” before the HEOM-QUICK2 calculation. The bath correlation functions by the Prony fitting spectrum decomposition scheme and truncate the hierarchy by the adiabatic scheme.

The input file is

1 6

2 4

3 0

4 1

5 2

6 2

7 5.0d0 5.0d0

8 0.4d0 0.4d0

9 0.001d0 0.001d0

10 0.001d0 0.001d0 -0.001d0 -0.001d0

11 2.5d2

12 5.d-3

13 $para1 eup=-1.0d0 edown=-1.0d0 uu=2.0d0 $end

14 $field fieldtype= 1 lreadomega = T $end

15 0.06d0 0.06d0 0.06d0 0.06d0

16 $jobinfo lsparse=T psfjob=T itype\_psf=1 $end

19 $adiabatic lad=T $end

20 $resume icont=0 lresume=T nresume=2000 $end

21 $dos ldos=T iorbs\_dos=1 ispin\_dos=1 lfreq\_dos=T

22 freq\_dos= xxxx maxit\_dos=20000 crit\_dos=1.d-7 $end

It is noted that the standard input file does not require the line numbers in the left side.

Users can run HEOM-QUICK2 with the command “./path/to/HEOM-QUICK2.x.x.x/bin/HEOM-QUICK2.x <input\_file\_name> out\_file\_name”.

These above examples are given in “/path/to/HEOM-QUICK2.x.x.x/readme/example”.

Now we introduce the tags in these input files.

* Line 1: task type;

“1”: time propagation from an initial state given by previous calculations;

“2”: iterative calculation to solve steady state from an initial state given by the program;

“3”: iterative calculation to solve steady state from an initial state given by previous calculations;

“4”: time propagation from an initial state given by the program;

“6”: calculate time-dependent response properties from “TAPE.resume”;

* Line 2: truncation tier;
* Line 3: the number of Padé or Matsubara poles, deactivated in Prony scheme;
* Line 4: the number of impurity;
* Line 5: spin degrees of freedom;
* Line 6: the number of baths;
* Line 7: band width of each bath;
* Line 8: coupling strengths between each bath and system;
* Line 9: temperature of each bath;
* Line 10: magnitude of spin-specific bias voltage;
* Line 11: length of time evolution, deactivated for steady-state calculation;
* Line 12: time step length;
* para1: the energetic parameters of SIAM;

The program offers other namelists for different model systems:

para2: two-level system;

para3: spinless two-impurity Anderson models;

para4: two-impurity Anderson models;

para5: three-impurity Anderson models;

para\_ hubbard: the single-site Hubbard model.

* field: external bias voltage:

fieldtype=0: exponential voltage, require the inverse of characteristic time;

fieldtype=1: sinusoidal voltage, require the frequency if lreadomega=T;

* jobinfo: the job control tag:

lsparse=T: use the sparse matrix technique;

psfjob =T: use the Prony fitting spectrum decomposition scheme;

psdfff =T: use the Fano spectrum decomposition scheme;

psdjob=T: use the Padé spectrum decomposition scheme, now default;

itype\_psf: the preset results of Prony scheme at different temperature;

itype\_fff: the preset results of Fano scheme with different scaling factors;

* converge: the control tag in iterative calculations;

maxit0: the maximum step in iteration calculations;

crit: the criterion in iteration calculations;

* method: the iterative algorithms in iterative calculations:

methodss=1: use the biconjugate gradient algorithm, does not support sparse matrix and the adiabatic truncation scheme;

methodss=2: use the transpose-free quasi-minimal-residue algorithm, supports sparse matrix and the adiabatic truncation scheme, now default;

methodss=4: use the Jacobi iteration algorithm, does not support the adiabatic truncation scheme;

* tdjob: the time propagation algorithms:

tdmethod=1: use the 4th-order Runge-Kutta algorithm, supports sparse matrix and the adiabatic truncation scheme, now default;

tdmethod=2: use the Chebyshev expansion algorithm, does not support sparse matrix and the adiabatic truncation scheme;

* adiabatic: the adiabatic truncation scheme: lad=T use the adiabatic scheme;
* resume: the continuation job control tags for time propagation:

icont=1 & lresume=T: resume from previous “TAPE.resume”;

nresume: save “TAPE\_(tt).resume” after running “nresume” steps;

* dos: the linear response calculation tags:

ldos: calculate linear response properties;

iorbs\_dos& ispin\_dos: the impurity and spin indices of system operators;

lfreq\_dos: calculate response properties in frequency domain based on the iterative method, now default;

freq\_dos: calculate response properties at a frequency equal to “freq\_dos”;

maxit\_dos: the maximum step in iteration calculations;

crit\_dos: the criterion in iteration calculations.

**License**

**HEOM-QUICK2 is licensed under the GNU General Public License v3.0**