A Quick Guide for Using my Generalized HEOM Code

Chang Woo Kim
January 1, 2025

1 Introduction and Disclaimer

This manual is written to share how to use my generalized HEOM code based on [Ikeda2020]. I created the initial version of this code in July 2020, when I was a postdoctoral researcher at Rochester. The code has been continuously extended and evolved ever since to fit my need as a reliable benchmark for method development in the field of open quantum system dynamics. However, due to the limited time for management, the structure of the code is far from optimal and there are some nearly deprecated components which cannot be guaranteed to work as I do not usually care about any backward compatibility.

For the reasons stated above, my aim is not to make this document a complete reference for all features offered by my code, but a minimal guide for conducting simulations of open quantum system dynamics and interpreting the results. As of today, the last major update of my code was done on October 2, 2024.

2 Installation

If you are outside of my group but still interested in using my code, send me an e-mail which contains a brief description of yourself and why HEOM is needed for your research. After being invited to access my private github repository, you can fetch the code by typing the following command:

```
git clone https://github.com/cwkim66/g-heom.git
```

The code is mostly based on the C language (and a bit of Fortran) and the compilation of the code is managed with Makefile. You will also need Intel MKL library for the installation, as it is the main workhorse for the linear algebra. The following environmental variables must be changed appropriately according to the structure of your system:

`FC`: The Fortran compiler.

`CC`: The C compiler.

`CPP`: The C++ compiler.

`MKLI`: Location of MKL header files.

`MKLL`: Location of MKL library files.

`GCCL`: Location of GCC library files.

All other variables should be kept as it is, unless you exactly know what you are doing.

Run `make` to compile the code and it will produce `g_heom.x` as the executable.

3 Options

You can run the code by feeding an option file to the executable:

```
./g_heom.x option
```

Of course, the location of the executable and the name of the option file can be changed.

All details about the simulation are contained in the option file. It consists of a number of keywords, each of which can be roughly classified into one of three categories depending on whether it:

- asks you to choose between multiple options

- accepts integer(s) as the argument

- accepts real number(s) as the argument

For the rest of the document, these three categories will be denoted by (mult), (int), and (real), respectively, to label a keyword according to its argument. All keywords are probably not case-sensitive, but I suggest following the naming conventions as much as possible to prevent any undesired outcomes.

3.1 General options

`$jtype` (mult): Determines the type of subsystem and the subsystem-bath interaction.

`0`: Electronic subsystem + spin-boson-type interaction.

`1`: Electronic subsystem + local interaction.

`2`: Electronic subsystem + arbitrary subsystem-bath interaction.

`10`: Vibronic subsystem + spin-boson-type interaction.

`11`: Vibronic subsystem + local interaction.

`12`: Vibronic subsystem + arbitrary subsystem-bath interaction.

`$nsite` (int): The dimensionality of the subsystem. Must be larger than 0.

`$initcond` (mult): Determines the type of initial condition for the subsystem reduced density matrix (RDM).

`site`: Localized excitation in the site basis.

`exci_fc`: Localized excitation in the exciton basis defined by the Franck-Condon excitation energy.

`exci_zero`: Same as `exci_fc` but the exciton basis is defined by the zero-phonon excitation energy.

`exci_fc_inco`: Incoherent mixture of the excitonic eigenstates whose populations are equal to a particular site basis. The exciton basis is defined by the Franck-Condon excitation energy.

`exci_zero_inco`: Same as `exci_fc_inco` but the exciton basis is defined by the zero-phonon excitation energy.

`custom`: Accepts an arbitrary RDM.

`$initexc` (int): Where the initial population will be placed, combined with the details provided by `$initcond`. Must be larger than 0 and lower or equal to `$nsite`.

Case 1: `$initcond` is either `site`, `exci_fc_inco`, or `exci_zero_inco`, description based on the site basis: Conventional labeling of the site basis, where 1 denotes the leftmost column of the Hamiltonian matrix.

Case 2: `$initcond` is either `exci_fc` or `exci_zero`, description based on the exciton basis: The exciton states are labeled with decreasing energy, with 1 denoting the state with the highest energy eigenvalue.

Case 3: `$initcond` is `custom`: Option will be ignored.

`$intgr` (mult): Integrator for time propagation.

`rk4`: Fourth-order Runge-Kutta (RK4) method with a fixed time step. Not very much used nowadays, due to the existence of more efficient RKF45 method.

`rkf45`: Fourth-order Runge-Kutta-Fehlberg (RKF45) method with adaptive time step. Currently recommended.

`$dt` (real): The size of the fixed time step when `$intgr = rk4`, and the initial guess for the appropriate time step when `$intgr = rkf45`.

`$dt_out` (real): The time step between the adjacent data points in the output files.

`$dt_max` (real): Maximum time step for time propagation when `$intgr = rkf45`. I found out that the simulation becomes inaccurate when the step size is solely controlled by the RKF45 integrator, and an upper limit should be imposed to retain the accuracy.

`$t_sim` (real): Total simulation time.

`$tol` (real): Error tolerance for the RKF45 integrator. The recommended value is 1.0×10^{-7}.

`$ndepth` (int): Depth of the HEOM hierarchy. Should be chosen to achieve balance between computational burden and convergence.

`$unit` (mult): Unit system for simulations.

`atomic`: Planck atomic units, where $\hbar = k_{\mathrm{B}} = 1$.

`waveno_fs`: Unit system where energy and time are measured in wavenumbers and femtoseconds, respectively.

`$prt_rho` (mult): Whether or not the code prints out the entire RDM as the output.

`0`: Do not print the RDM.

`1`: Print the RDM.

`$expansion` (mult): Decomposition scheme for the bath time correlation functions.

`matsu`: Matsubara decomposition originally introduced in [Tanimura1990]. Originally less favored due to the large number of terms required, but the utility was restored by an efficient low-temperature correction scheme reported in [Fay2022].

`pade`: Padé decomposition reported in [Hu2010]. Converges faster than Matsubara decomposition, and was the main decomposition scheme before the advent of [Fay2022]. Currently on the verge of deprecation in my code as it does not cope well with the correction scheme in [Fay2022].

`$ptb_lvl` (mult): Level of the low-temperature correction scheme according to [Fay2022].

`0`: No correction.

`1`: ???

`2`: Perturbative correction of the residual component of the bath correlation function after subtracting the infinite-temperature component. Currently recommended.

`3`: Currently broken. Originally planned to implement the perturbative correction for the truncated hierarchy beyond `$ndepth`, on top of the low-temperature correction offered by `$ptb_lvl = 2`.

`$Helec` (real): Hamiltonian matrix of the subsystem. The diagonal element measures the vertical (Franck-condon) excitation energies, and should not be confused with the zero-phonon excitation energies defined by the energy difference between PES minima.

`$rho_custom` (real): Subsystem RDM elements required when `$initcond` is `custom`. As the elements of the RDM are generally complex numbers, the user must provide (2 × `$nsite` × `$nsite`) real numbers as the input. Do not mark the imaginary part of a number as `I`, `j` or whatsoever, as the code only accepts a series of real numbers and inherently converts them as the complex number format.

3.2 Options for Bath Spectral Density

`$nspd` (int): Number of bath spectral densities (SPDs).

`$spd`: Marks the start of the SPD section. Each SPD is described by its own set of keywords, each of which starts with - instead of $. The SPD section is divided into subsections which correspond to individual SPDs. Each subsection must end with four hyphens (`----`), even when there is only one SPD. The keywords for defining the SPDs are listed below

`-type` (mult): Type of SPD function.

`drude`: Drude-Lorentz SPD.

`brosc`: Brownian oscillator SPD.

`-mode` (int): Determines the type of subsystem-bath interaction for the SPD. Relevant when `$jtype` is either `0`, `1`, `10`, and `11`, and ignored otherwise.

`0`: Spin-boson type coupling. Leads to error when `$nsite` is other than 2.

`n` ($1 \leq$ `n` $\leq$ `$nsite`): Local coupling to site `n`.

`-coup` (real): Coupling matrix for subsystem-bath interaction for the SPD. Relevant when `$jtype` is either `2` or `12`, and ignored otherwise.

`-temp` (real): Temperature of the SPD.

`-lambda` (real): Reorganization energy for both Drude-Lorentz and Brownian oscillator SPDs.

`-cutoff` (real): Cutoff parameter for both Drude-Lorentz and Brownian oscillator SPDs.

`-w0` (real): Center frequency for the Brownian oscillator SPD.

`-nmatsu` (int): Number of Matsubara functions explicitly used to construct the hierarchy. Warning: should be kept as `0` when you are relying on the perturbative correction scheme based on [Fay2022].

`-npade` (int): Number of Padé functions explicitly used to construct the hierarchy. Warning: should be kept as `0` when you are relying on the perturbative correction scheme based on [Fay2022].

`-ptb_corr` (int): Number of Matsubara or Padé functions that will be used for perturbative correction. The type of function is automatically determined by `$expansion`. Recommended value is `30` when the energy scale of the subsystem Hamiltonian is roughly close to the temperature of the bath.

`-nfhier` (int): Something related to Fano spectrum decomposition [Cui2019], which is required when the temperature of the bath is extremely low. Normally kept as `0`.

`-chi` (int): Similar to `-nfhier`. Normally kept as `1`.

`-thresh` (int) Something for preventing the divergence of the Matsubara/Padé decomposition when the characteristic frequency of a basis function becomes close to the cutoff frequency of the SPD. Almost deprecated, and I doubt it will function properly. Normally kept as `0.05`.

3.3 Options for Calling Vibronic Hamiltonian

Most of the keywords here are only relevant when `$jtype` is set as either `10`, `11`, or `12`, which demands vibronic Hamiltonian.

`$nqv` (int): The number of explicitly quantized bath modes. Should be kept as `0` when `$jtype` is other than `10`, `11`, or `12` or the simulation will not start.

`$qv_00_const` (mult): Deprecated option. Should be kept as `0` just in case.

`$qv_cutoff` (real): Sets the amount of residual thermal population, which is required to determine where we truncate the ladder of bosonic quantum states. `0.001` is often used for practical purposes, but can be increased for reducing the computational burden. Refer to `-nmax` for more information.

`$qvib` (int): Marks the start of the section for quantized vibration. Works similarly to `$spd` - the section is divided into subsections which correspond to individual quantized bath modes, and each subsection must end with four hyphens (`----`). Below are the keywords for describing each explicitly quantized mode.

`-siten` (int): Determines the type of subsystem-bath interaction for the explicitly quantized mode. Refer to `-mode` in Sec. 3.2, as the usage is identical.

`-coup` (real): Coupling matrix for subsystem-bath interaction for the explicitly quantized mode. Refer to `-coup` in Sec. 3.2, as the usage is identical.

`-nmax` (int): Number of bosonic quantum states used to describe the explicitly quantized mode.

`n` (`n` $\geq$ `1`): Use `n` bosonic quantum states.

`-1`: Automatically determine the number of bosonic quantum states based on the argument of `$qv_cutoff`. First, calculate the minimum value of N which satisfies

$$\frac{\sum_{k=0}^{N} \exp(-\beta E_k)}{\sum_{k=0}^{\infty} \exp(-\beta E_k)} > 1 - p,$$

where $\beta = (k_{\mathrm{B}}T)^{-1}$ is the inverse temperature, p is the argument of `$qv_cutoff`, and

$$E_k = \hbar\omega\left(k + \frac{1}{2}\right)$$

are the eigenenergies of the mode, with ω being its characteristic frequency. Namely, N is the minimum number of bosonic quantum states which captures the total population of $1 - p$, when the populations are distributed according to Boltzmann distribution. Having determined the value of N, the actual number of bosonic quantum states will be set as $\max(N, 5)$.

`-freq` (real): Characteristic frequency of the explicitly quantized mode.

`-hrfac` (real): Huang-Rhys factor associated with the explicitly quantized mode.

`-temp` (real): Temperature of the Boltzmann distribution, which affects `-nmax`.

4 Output Files

`exp_spin.txt`: The expectation value of Pauli spin operators $\hat{\sigma}_x$, $\hat{\sigma}_y$, and $\hat{\sigma}_z$, calculated by the RDM of the subsystem. Created only when `$nsite = 2`.

`pop.txt`: Electronic populations in the site basis.

`pop_exci.txt`: Electronic populations in the exciton basis. Not created when `$initcond` is `site`.

`ener.txt`: Energy expectation values of the electronic subsystem and explicitly quantized modes. Created only when `$jtype` is either `10`, `11`, or `12`.

5 References

[Tanimura1990] Tanimura, Y. Nonperturbative expansion method for a quantum system coupled to a harmonic-oscillator bath. *Phys. Rev. A* **1990**, *41*, 6676.

[Hu2010] Hu, J.; Xu, R-. X.; Yan, Y. Communication: Padé spectrum decomposition of Fermi function and Bose function. *J. Chem. Phys.* **2010**, *133*, 101106.

[Cui2019] Cui, L.; Zhang, H-. D.; Zheng, X.; Xu, R-. X.; Yan, Y. Highly efficient and accurate sum-over-poles expansion of Fermi and Bose functions at near zero temperatures: Fano spectrum decomposition scheme. *J. Chem. Phys.* **2019**, *151*, 024110.

[Ikeda2020] Ikeda, T.; Scholes, G. Generalization of the hierarchical equations of motion theory for efficient calculations with arbitrary correlation functions. *J. Chem. Phys.* **2020**, *152*, 204101.

[Fay2022] Fay, T. A simple improved low temperature correction for the hierarchical equations of motion. *J. Chem. Phys.* **2022**, *157*, 054108.

[Kim2022] Kim, C. W. Extracting bath information from open-quantum-system dynamics with the hierarchical equations-of-motion method. *Phys. Rev. A* **2022**, *106*, 042223.