QSLE v1.0
User Guide

Riccardo Cortivo, Mirco Zerbetto
Theoretical Chemistry Group – University of Padua – Italy

Index

1. Introduction

QSLE-v1.0 is the first program to simulate a system using the Quantum-Stochastic Liouville Equation. It has been created in 2024 by Riccardo Cortivo and Mirco Zerbetto of the TCG group of the University of Padua.

The program is based on:

1) Quantum-Stochastic Liouville Equation (QSLE) theory

2) Armadillo library : C++ Library for Linear Algebra & Scientific Computing

3) spline.h : a simple cubic spline interpolation library without external dependencies

Please see Section 2 for all the related papers.

2. Citation Details

Please cite the following papers if you use QSLE software in your research. Citations are useful for the continued development and maintenance of the libraries.

- Jonathan Campeggio, Riccardo Cortivo and Mirco Zerbetto, "A multiscale approach to coupled nuclear and electronic dynamics. I. Quantum-stochastic Liouville equation in natural internal coordinates," J. Chem. Phys., **158**, 244104 (2023)
- Riccardo Cortivo, Jonathan Campeggio and Mirco Zerbetto, "A multiscale approach to coupled nuclear and electronic dynamics. II. Exact and approximated evaluation of nonradiative transition rates," J. Chem. Phys., **158**, 244105 (2023)
- Conrad Sanderson and Ryan Curtin, "Armadillo: a template-based C++ library for linear algebra," J. Open Source Softw., **1**, *2*, 26 (2016)
- Conrad Sanderson and Ryan Curtin, "Practical Sparse Matrices in C++ with Hybrid Storage and Template-Based Expression Optimisation," Math. Comput. Appl., **24**, *3* (2019)

All the information regarding the implementation of this software will be inserted into an upcoming paper.

3. Installation

QSLE-v1.0 software can be installed only on Linux systems via CMake, with or without root access. Before compiling the libraries, please read the following subsection, in which the minimum requirements are listed.

3.1 Prerequisites

Before installing QSLE-v1.0 software, please check if the system meets the following requirements:

- C++11 (that is fully supported by GCC 4.8+ and Clang 3.4+)
- CMake 3.1+
- OpenMP 3.1+
- LAPACK shared library and its development files
- BLAS (or OPENBLAS) shared library and its development files
- ARPACK shared library and its development files
- SuperLU shared library version 5.2.x, 5.3.x, or 6.0.x (6.0.x must be compiled with default integer size (32 bits)) and its development files

For example, in Ubuntu & Debian distribution, you need to install:

```
sudo apt-get install gcc g++ clang cmake -y
sudo apt install libopenblas-dev liblapack-dev libarpack2-dev libsuperlu-dev -y
```

3.2 How to compile QSLE-v1.0 software from terminal

To install QSLE-v1.0, just unpack the tgz package file

```
tar zxf QSLE-v1.0.tgz
```

step into the `QSLE-v1.0` folder and run the `build.sh` script

```
cd QSLE-v1.0
./build.sh
```

following the instructions printed on screen (type 1 and press enter). The script will first compile the zmatlib and DiTe2lib libraries located in the lib sub-folder, and then will build the parallel standalone version of QSLE-v1.0. The generated executable `qsle` can be found in the `QSLE-v1.0` directory, in order to make it easier to access (we suggest adding the `QSLE-v1.0` directory to the path, but it is not mandatory).In addition, the `run_qsle.sh` bash script will be generated in the `QSLE-v1.0` directory, that is an example of how to run the program using more cores.

3.3 Parallel support

QSLE-v1.0 uses OpenMP to speed up computationally expensive operations. Before starting the run, you must set the following environment variable in your terminal or bash script

```
export OMP_NUM_THREADS=N
```

where N is the number of cores you want to use.
N.B.: To use OpenMP in a cluster node, you have to follow the instructions of the job scheduler. For example, in SLURM job scripts you have to set the following keywords

```
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=N
```

where N is the number of cores you want to request.

MPI version of the software has not been implemented yet. For this reason, the program can be run only in a single node.

4. How to run the software from terminal

After the compiling procedure, the software can be used following three steps:

1) Writing the input file (see Section 4.1)

2) Setting OMP_NUM_THREADS and QSLEINFO environment variable

```
export OMP_NUM_THREADS=N
export QSLEINFO=/path/to/software/directory/QSLE-v1.0/src/qsle_info
```

where N is the number of cores you want to request, and path/to/software/directory/ has to be substituted with the path preceding QSLE-v1.0 folder

3) Run the appropriate executable in the terminal followed by the input file

```
/path/to/software/directory/QSLE-v1.0/qsle QSLE_INPUT_FILE
```

where QSLE_INPUT_FILE have to be substituted with the real name of the input file.

N.B.: If you want to redirect the terminal output to a log file, just add >file.log after the name of the input.

If you want to run the software in your local machine, you can copy and use in your working folder run_qsle.sh bash scripts, where you just need to change the name of QSLE_INPUT_FILE (and the number of cores needed).

4.1 Input structure

The general structure of the input is

```
…
KeYwOrDx  Argument_of_keywordX  comments_by user
kEyWoRdY    Argument_of_keywordY
…
```

and it is characterized by these rules:

- the keywords are case unsensitive
- the arguments of the keywords are case sensitive (except for some specific cases)
- the keywords must be separated from their arguments by white spaces (or tab spaces)
- the comments of the user can be placed after the arguments (provided there is at least a white space separating them)

The list of keywords is here reported:

- Prefix : string that will be used as a prefix for all the output of the program. The default is QSLE.
- Restart : double precision number that corresponds to the last output probability density that has been written by the program.

  N.B.: if you have already computed the evolution matrix related to the system (see Section 5), then you can set this number to 0 in order to restart your run. This is useful to restart the QSLE evolution with a new initial condition or with a new timestep, since these two features do not affect the evolution matrix.

- `Pes`: name of the file containing the Potential Energy Surfaces (PESs). The file must reflect the following structure

```
-3.14      -100      -98
-3.13      -99.999   -97.999
…          …         …
3.13       -99.998   -97.998
3.14       -100      -98
```

where the first column is the torsional angle (φ) in radians (rad) using ascending order, the second column corresponds to the energy of the Ground State (GS, or state 0) in electronVolts (eV), and the third column is the energy of the Excited State (ES, or state 1) in eV.

N.B.: the domain of the torsional angle must be [-π,π] (extremes included, which means that the energies at -π must be equal to the energies at π), and π must be approximated at least with 3.14. The spacing between two consecutive points can be irregular, but huge gaps are not recommended, otherwise the fitting procedure can produce bad results.

- `d01` : the name of the file containing the derivative coupling vector $\langle \psi_0 | \partial / \partial \varphi | \psi_1 \rangle$ between the ground and the excited state. The file must reflect the following structure

```
-0.5       0.001
-0.4       0.005
…          …
0.2        0.004
0.3        0.002
```

where the first column is the torsional angle in radians using ascending order (at least 3 values), and the second column corresponds to the derivative coupling in rad^{-1}.

N.B.: there is no mandatory domain for the torsional angle. The spacing between two consecutive points can be irregular, but huge gaps are not recommended, otherwise the fitting procedure can produce bad results.

TIP: if the nonadiabatic event takes place in limited zones, you can just compute the derivative coupling vector in these zones and fill the gaps between them with zeros.

- `mu01` : the name of the file containing the dipole momentum between the ground and the excited state as a function of the dihedral angle. The file must reflect the following structure

```
-3.14      1
-3.13      1.1
…          …
3.13       1.1
3.14       1
```

where the first column is the torsional angle in radians using ascending order, and the second column corresponds to the dipole moment in Debye.

N.B.: the dipole moment must be defined for every torsional angle. The spacing between two consecutive points can be irregular, but huge gaps are not recommended, otherwise the fitting procedure can produce bad results.

- wrad : the name of the file containing the extra radiative relaxation rates between the ground and the excited state as a function of the dihedral angle. The file must reflect the following structure

```
-3.14      0
-3.13      0.1
…          …
3.13       0.1
3.14       0
```

where the first column is the torsional angle in radians using ascending order, and the second column corresponds to relaxation rates in femtoseconds.

N.B.: the dipole moment must be defined for every torsional angle. The spacing between two consecutive points can be irregular, but huge gaps are not recommended, otherwise the fitting procedure can produce bad results.

- wnorad : the name of the file containing the extra nonradiative relaxation rates between the ground and the excited state as a function of the dihedral angle and the momentum. The file must reflect the following structure

```
-0.5     -1       0
-0.4     -0.9     0.1
…        …        …
0.4      0.9      0.1
0.5      1        0
```

where the first column is the torsional angle in radians using ascending order, the second column is the momentum in amu*Å^2*rad/fs, and the third column corresponds to nonradiative relaxation rates in femtoseconds.

N.B.: there is no mandatory domain both for the torsional angle and for the momentum. The spacing between two consecutive points can be irregular, but huge gaps are not recommended, otherwise the fitting procedure can produce bad results.

- Pdb : name of the Protein Data Bank (PDB) file containing the molecule structure. For other information, please refer to the documentation of DiTe2 software.
- Refatoms : IDs of the four reference atoms constituting the dihedral angle (the numbering is based on the original PDB file). This argument is characterized by four integer numbers, for example

```
Refatoms 17 22 15 14
```

For other information, please refer to the documentation of DiTe2 software.

- Angle_points : integer number corresponding to the number of points on the axis of the torsional angle (default 300).
- Momentum_points : integer number corresponding to the number of points on the axis of the momentum associated to the torsional angle. It must be odd (default 301).
- Derivation_method : it represents how $\partial/\partial\varphi$ and $\partial/\partial p$ are computed, and it takes as an argument only
  - ➔ FD stands for Finite Difference method (default)
  - ➔ LRBF stands for Local Radial Basis Function method, a more refined approach that requires two additional keywords
    - o Angle_domain N1
    - o Momentum domain N2

    Where N1 is the integer number that represent the number of points of the local angle domain (default 15), while N2 is the number of points of the local momentum domain (default 15).
- Timestep : double precision number corresponding to the timestep of the QSLE evolution in femtoseconds (fs) (default 0.005).
- Total_steps : integer number corresponding to the total steps to be computed (default 200000000)
- Restart_step : integer number indicating in how many steps the probability is printed out (default 20000)
- Normalization_step : integer number indicating in how many steps the probability is normalized out (by default it is equal to Restart_step)
- Temperature : temperature of the system in Kelvin (default 300).
- Ek_max : double precision number corresponding to the maximum value of the kinetic energy of the system in k_BT, where k_B is the Boltzmann constant, and T is the temperature (default 10.0)
- Mom_max : double precision number corresponding to the maximum value of the momentum (equal to the absolute value of the minimum) in amu*Å^2*rad/fs. If this keyword is set, Ek_max is ignored.
- Friction : double precision number corresponding to the friction experienced by the dihedral angle in SI units (J*s).

  N.B.: By default, the friction is computed using DiTe2 software, but you can force the system to experience a fictitious friction using this keyword.
- Decoherence_time : double precision number corresponding to the decoherence time of QSLE transition rates in fs.

  N.B.: if not provided, the decoherence time is considered equal to the timescale of loss of self-correlation of the particle velocity for free diffusion.
- Initial_distribution : the keyword is related to the initial distribution of the system. It takes as an argument only
  - ➔ Boltzmann if you want to start from Boltzmann probability distribution
  - ➔ Dirac A0 M0 if you want to start from a probability distribution centered on the specific angle A0 and the specific momentum M0 (A0 M0 are double precision numbers)
  - ➔ filename.dat if you want to start from a custom probability density (here the argument is case sensitive)

    N.B.: The structure of the custom probability density must be equal to the structure of the output probability density (see Section 5)

- `Force_transition` : the keyword can transfer the population of the initial distribution of the system from ground to excited state or vice versa. It takes as an argument only
    - ➔ `None` if you don’t want to influence the initial distribution (default)
    - ➔ `Up` if you want to move all the probability density of the ground state to the excited state
    - ➔ `Down` if you want to move all the probability density of the excited state to the ground state
- `Reff` : double precision number associated to hydrodynamic effective beads radius (default `2.0`). For other information, please refer to the documentation of DiTe2 software.
- `C` : double precision number associated to hydrodynamic boundary conditions (default `6.0`). For other information, please refer to the documentation of DiTe2 software.
- `Viscosity` : double precision number associated to hydrodynamic viscosity in Pa*s (default `8.9e-4` , that is the water viscosity at 298.15 K). For other information, please refer to the documentation of DiTe2 software.

5. Output

During a run, QSLE-v1.0 will print the following output (Prefix is substituted with the string associated to that keyword):

- Prefix_g0.dat : the covariant coefficient g0 (that is the inertia of the dihedral angle in amu*Å^2) is printed as a function of the dihedral angle in rad.
- Prefix_m01.dat : the transition rates from excited state to ground state in fs^{-1} are printed as a function of the dihedral angle in rad and the momentum in amu*Å^2*rad/fs.
- Prefix_m10.dat : the transition rates from ground state to excited state in fs^{-1} are printed as a function of the dihedral angle in rad and the momentum in amu*Å^2*rad/fs.
- Prefix_matrix : the evolution matrix to propagate the QSLE is stored in armadillo compressed format (not readable).
- Prefix_restart.inp : a simple file that contains all the information needed to restart the run from the last output printed.
- Prefix_DiTe2_diffusion.dat : diffusion tensor associated to the system (see DiTe2 manual for further details)
- Prefix_DiTe2_friction.dat : friction tensor associated to the system (see DiTe2 manual for further details)
- Prefix_time_t.dat : the probability density at time t (time is expressed fs), that depends on the dihedral angle in rad and the momentum in amu*Å^2*rad/fs.

  N.B.: this output is not easily readable, for this reason there are some Python tools that can make the output easier to understand (see Section 6).

  N.B.: the probability density is normalized using the units of the program (rad, fs, amu ecc.). If you want to modify the units of the output, please read Section 6.

6. Python tools

QSLE-v1.0 software offers some jupyter notebooks useful to process the output after the run. These notebooks can be found in the `python_tools` folder and require some specific packages such as numpy, scipy ecc. (they are listed inside each notebook).

In this section, a brief description of each jupyter notebook is provided. For the detailed instructions, please carefully read each notebook.

6.1 Output_Converter.ipynp

This jupyter notebook can be used to convert the units of measure of the computed probability densities (i.e. the `Prefix_time_t.dat` output files). For example, you can change from radians to degrees, or from femtoseconds to picoseconds, in order to make the output easier to understand.

In addition, the converted probability density is printed in a gnuplot readable format, which is easier to read and to use for further processing.

All the information about how to use this jupyter notebook and which is the structure of the new output is written inside the `Output_Converter.ipynb` file.

6.2 Plot_Probability_Densities.ipynp

This jupyter notebook can be used to generate the surface plot of the computed probability densities (i.e. the `Prefix_time_t.dat` output files), using the set of units of your choice. In addition, the percentages of the population of the ground state and the excited is displayed.

You can find all the information about how to use this jupyter notebook in the `Plot_Probability_Densities.ipynb` file.

6.3 Plot_Transition_Coefficients.ipynb

This jupyter notebook can be used to generate the surface plot of the computed transition coefficients (i.e. the `Prefix_m01.dat` and `Prefix_m10.dat` output files), using the set of units of your choice.

You can find all the information about how to use this jupyter notebook in the `Plot_Probability_Densities.ipynb` file.

6.4 Spectra_Generator.ipynb

This jupyter notebook can be used to generate the spectra based on the computed probability densities (i.e. the `Prefix_time_t.dat` output files), using the set of units of your choice.

You can find all the information about how to use this jupyter notebook in the `Plot_Probability_Densities.ipynb` file.

N.B.: The intensities of the spectra are based only on the probability densities and the energy gap between the states. Changing in the absorption/emission coefficient based at different dihedral angles are not taken into account.

7. Example

You can find a simple test for the QSLE-v1.0 software in the `example` folder. In particular, you can find all the files that are needed to simulate a simple molecule based on four atoms:

- `4bead_hexane.inp` ⎵ input file
- `4bead.pdb` ⎵ pdb structure file
- `4bead_adiabatic_PESs.dat` ⎵ molecule potential energy surfaces
- `4bead_d01.dat` ⎵ nonadiabatic coupling vector associated to the molecule

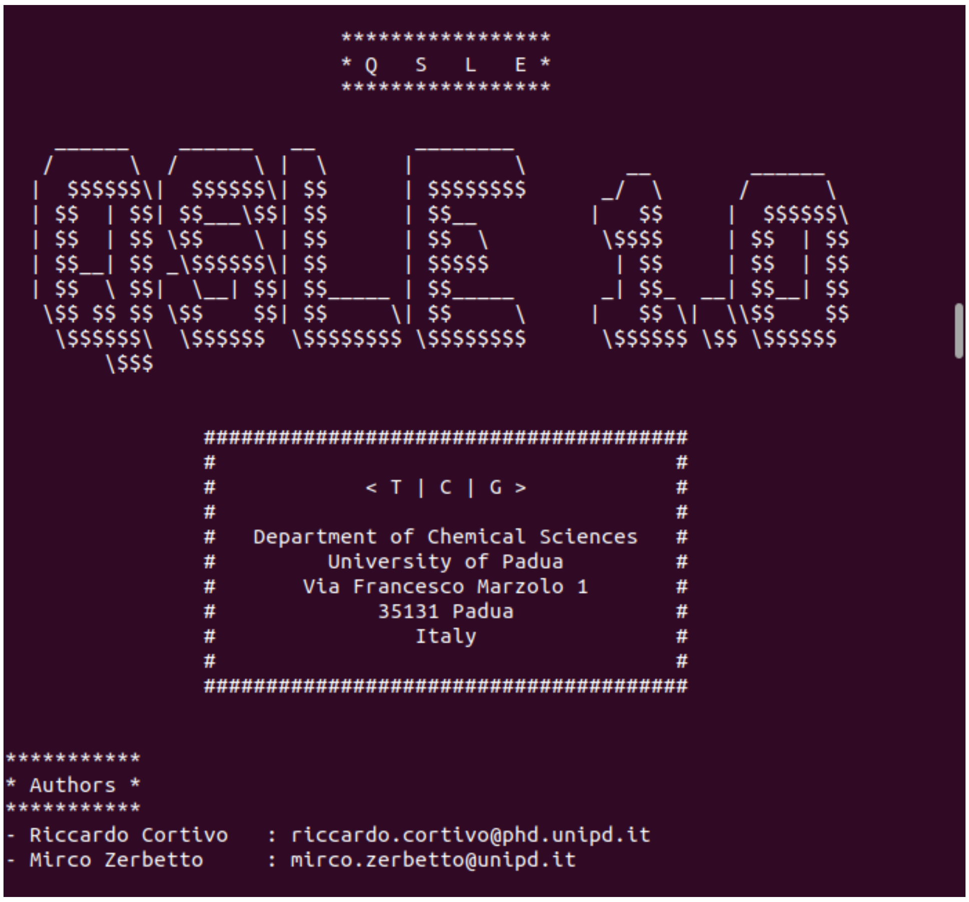
The system described is not a real one, since the atom type M used in the pdb file is not a real atom type. All the information related to the example system are reported in an upcoming article.

To test the software after having installed it, please follow these instructions:

1) Copy the `run_qsle_serial.sh` bash script in the `example` folder
2) Go into the `example` folder, and open the `run_qsle_serial.sh` bash script
3) Substitute `QSLE_INPUT_FILE` with `4bead_hexane.inp`
4) Open the terminal and type

   ```
   bash run_qsle_serial.sh
   ```

   The run should start, and you should see an output like this:



```
                           *****************
                           * Q   S   L   E *
                           *****************


    ______    ______   __        ________
   /      \  /      \ |  \      |        \      __      ______
  |  $$$$$$\|  $$$$$$\| $$      | $$$$$$$$$     _/  \     /      \
  | $$  | $$| $$___\$$| $$      | $$__        |   $$     |  $$$$$$\
  | $$  | $$ \$$    \ | $$      | $$  \        \$$$$     | $$  | $$
  | $$__| $$ _\$$$$$$\| $$      | $$$$$          | $$     | $$  | $$
  | $$  \ $$|  \__| $$| $$_____ | $$_____       _| $$_  __| $$__| $$
   \$$ $$ $$ \$$    $$| $$     \| $$     \     |   $$ \|  \\$$    $$
    \$$$$$$\  \$$$$$$  \$$$$$$$$ \$$$$$$$$      \$$$$$$ \$$ \$$$$$$
        \$$$


                ########################################
                #                                      #
                #            < T | C | G >             #
                #                                      #
                #   Department of Chemical Sciences    #
                #         University of Padua          #
                #       Via Francesco Marzolo 1        #
                #             35131 Padua              #
                #                Italy                 #
                #                                      #
                ########################################


***********
* Authors *
***********
- Riccardo Cortivo   : riccardo.cortivo@phd.unipd.it
- Mirco Zerbetto     : mirco.zerbetto@unipd.it
```

If this happens, the software should be installed correctly, and you can start a new run with the system of your choice!