THE SCINE SPARROW DEVELOPERS:

FRANCESCO BOSIA, TAMARA HUSCH, ALAIN VAUCHER, AND MARKUS REIHER

USER MANUAL

SCINE SPARROW 2.0.1

ETH ZÜRICH

Contents

Introduction

The availability of fast electronic energies and gradients is essential for the SCINE project. The SCINE Sparrow module contains electronic structure models which were designed to yield electronic energies, energy gradients with respect to the nuclear coordinates, and Hessians rapidly. The SCINE Sparrow module can be driven from SCINE Interactive, SCINE ReaDuct, and SCINE Chemoton. However, as with all SCINE modules it is also a stand-alone program which can be applied on its own or easily interfaced to other programs.

SCINE Sparrow is a command-line tool that implements many popular semiempirical models. SCINE Sparrow 2.0.1 provides the `MNDO`, `AM1`, `RM1`, `PM3`, `PM6`, non-SCC DFTB (`DFTB0`), `DFTB2`, and `DFTB3` methods (open- and closed-shell formalisms are implemented). The application of semiempirical models usually allows for rapid calculation of electronic energies and energy gradients for a small molecular structure with a given charge and spin state.

In this manual, we describe the installation of the software, an example calculation as a hands-on introduction to the program, and the most import functions and options.[1] A prospect on features in future releases and references for further reading are added at the end of this manual.

[1] Throughout this manual, the most import information is displayed in the main text, whereas useful additional information is given as a side note like this one.

Obtaining the Software

Sparrow is distributed as open source software in the framework of the SCINE project (www.scine.ethz.ch). Visit our website (www.scine.ethz.ch/download/sparrow) to obtain the software.

System Requirements

Sparrow itself has only modest requirements regarding the hardware performance. However, the underlying quantum-chemical calculations might become resource intensive if extremely large systems are studied.

Installation

Sparrow is distributed as an open source code. In order to compile Sparrow from this source code, you need

- A C++ compiler supporting the C++14 standard (we recommend gcc 7.3.0),
- cmake (at least version 3.9.0),
- the Boost library (we recommend version 1.64.0), and
- the Eigen3 library (we recommend version 3.3.2).

In order to compile the software, either directly clone the repository with git or extract the downloaded tarball, change to the source directory and execute the following steps:

```
git submodule init
git submodule update
mkdir build install
cd build
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=../install ..
make
make test
make install
export PATH=$PATH:<source code directory>/install/bin
```

This will configure everything, compile your software, run the tests, and install the software into the folder "install". Finally, it will add the Sparrow binary to your `PATH`, such that you can use it without having to specify its full location. In this last command, you have to replace `<source code directory>` with the full path where you stored the source code of Sparrow.

In case you need support with the setup of Sparrow, please contact us by writing to scine@phys.chem.ethz.ch.

Example Calculation

Sparrow is a command-line-only binary; there is no graphical user interface. Therefore, you always work with the Sparrow binary on a command line such as the Gnome Terminal or KDE Konsole. All functionality is accessed via command line arguments. All possible command line options can be listed with the following command:

```
sparrow --help
```

In order to provide a practical demonstration of the Sparrow program, we present here a step-by-step example calculation that guides you through the complete process of calculating the total electronic energy as well as the nuclear gradient and Hessian of a molecular structure with Sparrow. We start with the following Cartesian coordinates for water:

```
3

O      -0.27939703       0.83823215       0.00973345
H      -0.52040310       1.77677325       0.21391146
H       0.54473632       0.90669722      -0.53501306
```

Store these coordinates in a file named "h2o.xyz". Then, call Sparrow with the following command:

```
sparrow  --structure h2o.xyz --molecular_charge 0 --spin_multiplicity 1 --method PM6
```

This will calculate the electronic energy for the neutral water molecule with PM6. You can also use the short options

```
sparrow -x h2o.xyz -c 0 -s 1 -M PM6
```

to achieve the same result. If you also want to calculate the nuclear gradient, simply add the option `--gradient` or `-G`. For the Hessian, specify `--hessian` or `-H`.

Detailed Documentation

Command Line Arguments

In this section, the full functionality of Sparrow is documented, *i.e.*, all possible command line arguments are listed and explained.

- `--structure`, `-x`: This argument specifies the structure which should be calculated. It must be given as a path to an XYZ file.
- `--molecular_charge`, `-c`: This is used to specify the overall charge of the system to be calculated, `e.g.`, `--molecular_charge 0` or `-c -1`. The default charge is zero. Only charges between -20 and + 20 are supported.
- `--spin_multiplicity`, `-s`: This is used to specify the spin multiplicity of the system to be calculated, `e.g.`, `-s 1` for a singlet state. The default multiplicity is one. The maximal spin multiplicity is 10.
- `--gradients`, `-G`: If given, the nuclear gradients will be calculated.
- `--hessian`, `-H`: If given, the Hessian and the nuclear gradients will be calculated.
- `--thermochemistry`, `-C`: If given, the heat capacities at constant volume or pressure, the enthalpy, the entropy, the Gibbs free enthalpy as well as the zero point vibrational energy (ZPVE) are calculated from the total molecular partition function. The calculation of the thermochemical properties requires the Hessian matrix: its calculation is therefore implied by setting this option.
- `--temperature`, `-T`: This is used to specify the temperature in Kelvin at which the thermochemical properties are calculated. The default temperature is 298.15 K. The maximum allowed temperature is 10'000 K.
- `--suppress_normal_modes`, `-N`: If given, the full Hessian will be

printed instead of the normal modes and the vibrational frequencies. This option has no effect if the Hessian is not calculated (see above).

- `--bond_orders`, `-B`: If given, the bond order matrix will be calculated.
- `--method`, `-M`: With this option, the desired calculation method can be set. Options in SPARROW 2.0.1 are `MNDO`, `AM1`, `RM1`, `PM3`, `PM6`, `DFTB0`, `DFTB2`, and `DFTB3`. By default, `PM6` is selected.
- `--output_to_file`, `-o`: If this option is given, the output will not only be printed to the screen, but also to files. By default, the energy is stored in a file named "energy.dat", the nuclear gradients in a file named "gradients.dat", the Hessian in a file named "hessian.dat", and the bond order matrix in a file named "bond_orders.dat". If a description is given with the option `-d` (see below), this description will also be used in the file name.
- `--description`, `-d`: This can be used to add a (short) description of the calculation. This description will appear in the output. This allows to quickly find a certain calculation output at a later point. The description should be enclosed by quotation marks if it is composed by more than one word, *e.g.*, `-d "This is an example"`.
- `--unrestricted_calculation`, `-u`: If this option is given, an unrestricted (UHF) calculation will be performed. By default, a restricted calculation will be done.
- `--wavefunction`, `-W`: If this option is given, a Molden input file is generated after the calculation to allow for the visualization of the molecular orbitals.The basis functions used for the generation of the Molden input are defined in the files `<source code directory>/Sparrow/Sparrow/Resources/<method>/STO-6G.basis`. All `NDDO` methods have their own STO-6G expansion which are fine tuned based on the method's parameters. For the `DFTB` methods, the same basis set as `PM6` is used.

The following options are usually not needed:

- `--help`, `-h`: This prints a short help message listing and explaining all possible command line arguments
- `--scf_mixer`, `-m`: With this option, the method used to accelerate the convergence of the self- consistent-field (SCF) calculations can be set. Possible options are 'no_mixer' (no convergence acceleration), 'diis' (direct inversion of the iterative subspace, DIIS), 'ediis' (energy DIIS), and 'ediis_diis'. The default is 'diis'.

- `--max_scf_iterations`, `-i`: This is used to specify the maximum number of SCF iterations. Default is 100.
- `--self_consistence_criterion`, `-t`: This specifies the convergence threshold for the electronic energy. This value is given in hartree. By default, it is '1e-5'.
- `--parameter_file`, `-p`: This option can be used to specify the path of the parameter file to be used. This option is usually not needed, since SPARROW provides default parameter files for all its methods.
- `--parameter_root`, `-P`: This option can be used to specify a directory in which SPARROW should search for its parameter files. This option is usually not needed, since SPARROW provides default parameter files for all its methods. The path to the parameter file (or directory for the DFTB methods) is the concatenation of the `parameter_root` and `parameter_file` options.
- `--log`, `-l`: This sets the log level for warning messages and errors. Supported levels are `trace`, `info`, `warning`, `error`, and `fatal`. By default, the level is set to `info`, *i.e.*, all warnings and errors are printed to STDERR. If you set this option to `error`, only errors are printed. If you set this to `none`, neither warnings nor errors are printed.

Running SPARROW *in Parallel*

By default, SPARROW will be compiled with OPENMP support and hence, it can be run in parallel. In order to use multiple CPU cores, simply specify

```
export OMP_NUM_THREADS=n
```

where `n` is the number of CPUs you want to use. Note that by default, SPARROW uses all available cores, *i.e.*, it will also run in parallel if you do not specify the above environment variable.

Using the Python Bindings

Sparrow provides Python bindings such that all functionality of Sparrow can be accessed also via the Python programming language. In order to build the Python bindings, you need to specify `-DSCINE_BUILD_PYTHON_BINDINGS=ON` when running cmake (see also chapter Installation).

In order to use the Python bindings, you need to specify the path to the Python library in the environment variable `PYTHONPATH`. Additionally, the path to the Sparrow module needs to be added to the environment variables `SCINE_MODULE_PATH`, *e.g.*, you have to run the command

```
export PYTHONPATH=$PYTHONPATH:<source code directory>/install/lib/python<version>/site-packages
export SCINE_MODULE_PATH=<source code directory>/install/lib
```

where `<version>` is the Python version you are using (*e.g.*, 3.6). Now, you can simply import the library and use it as any other Python library. For example, in order to calculate the total electronic energy of H_2 with the `AM1` method, use the following Python statements:

```
import scine_sparrow

calculation = scine_sparrow.Calculation('AM1')
calculation.set_elements(['H', 'H'])
calculation.set_positions([[0, 0, 0], [1, 0, 0]])
calculation.calculate_energy()
```

Note that the atomic coordinates are specified in Å (*i.e.*, basically, the XYZ format is used). The output of Sparrow is always in Hartree atomic units.

A detailed list of all the functions provided by the Sparrow Python library can be found by running

```
import scine_sparrow
```

```
help(scine_sparrow)
```

Extensions Planned in Future Releases

- Availability of OMx models
- Availability of the CISE approach
- Calculation of excited states
- Implementation of periodic boundary conditions

References

Please consult the following references for more details on Sparrow. We kindly ask you to cite the appropriate references in any publication of results obtained with Sparrow.

- Primary reference for Sparrow 2.0.1: F. Bosia, T. Husch, A. C. Vaucher, M. Reiher, "qcscine/sparrow: Release 2.0.1 (Version 2.0.1)", Zenodo, 2020.
- Presentation of the formalism of MNDO-type and OMx models:

  T. Husch, A. C. Vaucher, M. Reiher "Semiempirical Molecular Orbital Models Based on the Neglect of Diatomic Differential Overlap Approximation", *Int. J. Quantum Chem.*, **2018**, *118*, e25799.
- Presentation of DFTB approaches:

  M. Elstner, G. Seifert, "Density functional tight binding", *Phil. Trans. R. Soc. A*, **2014**, *371*, 20120483.
- Presentation of CISE:

  T. Husch, M. Reiher "Comprehensive Analysis of the Neglect of Diatomic Differential Overlap Approximation", *J. Chem. Theory Comput.*, **2018**, *14*, 5169.