sophisticated-dmrg Documentation

Release 1.0

James R. Garrison and Ryan V. Mishmash

August 05, 2014

Contents

Source code: https://github.com/simple-dmrg/sophisticated-dmrg/

Documentation: http://sophisticated-dmrg.readthedocs.org/

This code is an expanded density-matrix renormalization group (DMRG) program, based on code written for a tutorial given originally at the 2013 summer school on quantum spin liquids, in Trieste, Italy. It implements DMRG its traditional formulation (i.e. without using matrix product states). DMRG is a numerical method that allows for the efficient simulation of quantum model Hamiltonians. Since it is a low-entanglement approximation, it often works quite well for one-dimensional systems, giving results that are nearly exact.

Typical implementations of DMRG in C++ or Fortran can be tens of thousands of lines long. Here, we have attempted to strike a balance between clear, simple code, and including many features and optimizations that would exist in a production code. One thing that helps with this is the use of Python. We have tried to write the code in a very explicit style, hoping that it will be (mostly) understandable to somebody new to Python.

CHAPTER 1

Features

Beyond the features already existing in simple-dmrg (infinite and finite system algorithms, conserved abelian quantum numbers, and eigenstate prediction), sophisticated-dmrg offers the following improvements:

- pluggable models
  - Heisenberg XXZ
  - Bose-Hubbard
- choice between open or periodic boundary conditions
- measurements (assumes operators on different sites commute)
- site-dependent potential (e.g. to implement disorder)

CHAPTER 2

Future features

2.1 Planned and potential features

- use disk (not RAM) for persistent storage
- efficient representation of the Hamiltonian (if easily possible in python)
- time-dependent DMRG
- custom Lanczos
- fermions and fermionic Hubbard models
- models for ladder systems
- site-dependent hopping terms (e.g. to implement “hopping disorder”)

2.2 Highly unlikely future features

- rewrite in terms of matrix product states
- non-abelian symmetries (e.g. SU(2))

CHAPTER 3

Authors

- James R. Garrison (UCSB)
- Ryan V. Mishmash (UCSB)

CHAPTER 4

Contents

4.1 Using the code

The requirements are:

- Python 2.6 or higher (Python 3 works as well)
- numpy and scipy

Download the code using the Download ZIP button on github, or run the following command from a terminal:

```
$ wget -O sophisticated-dmrg-master.zip https://github.com/simple-dmrg/sophisticated-dmrg/archive/mas
```

Within a terminal, execute the following to unpack the code:

```
$ unzip sophisticated-dmrg-master.zip
$ cd sophisticated-dmrg-master/
```

Once the relevant software is installed, each program is contained entirely in a single file. The first program, for instance, can be run by issuing:

```
$ python sophisticated_dmrg.py
```

Note: If you see an error that looks like this:

```
SyntaxError: future feature print_function is not defined
```

then you are using a version of Python below 2.6. Although it would be best to upgrade, it may be possible to make the code work on Python versions below 2.6 without much trouble.
