ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computational Quantum Physics
Exercise 4

FS 09
Prof. P. Werner,
Prof. P. De Forcrand

Submitting Block B You should submit your results for Block B, i.e. the exercises 2.1, 2.2, 3.1 and if you want the bonus exercises. Please submit the source code together with your results by the evening of Sunday, March 29, 2009 to one of the assistants.

Problem 4.1 Finite-size DMRG

This exercise is meant as a bonus exercise for those who have finished their infinite-size DMRG program. Based on this, you can now write a finite-size DMRG by implementing the sweeping procedure as described in the script (or the review, see below). To this end, you need to

1. keep track of all block Hamiltonians and operators that you calculate,
2. extend your update procedure to be able to handle blocks of different lengths.

With this, it should be straightforward to add the sweeping. You can measure how the energy changes with the number of sweeps and you should observe a decrease.

Problem 4.2 Speeding up your calculation

This exercise is also meant as a bonus exercise. It is probably much easier to do in Matlab than in C++.
The superblock Hamiltonians will usually have a sparse structure that can be exploited to reduce the complexity of the algorithm from $\mathcal{O}(M^6)$ to $\mathcal{O}(M^4)$ by using iterative eigenvalue solvers that rely only on matrix-vector products instead of matrix-matrix products, such as the Lanczos method. You can find these implemented in Matlab or in C++ libraries such as IETL (http://www.comp-phys.org/software/ietl/). Try to use one of these methods in your code.
There are other tricks that further reduce the complexity by never constructing the full superblock Hamiltonian matrix. You can find these described in a review by U. Schollwoeck, Rev. Mod. Phys. 77, 259 (2005). It is also available from http://arxiv.org/abs/cond-mat/0409292.