Ji-Woo Lee

Numerical Methods for quantum systems

THE FIRST EDITION

KIAS/APCTP

PUBLISHED BY KIAS/APCTP

TUFTE-LATEX.GOOGLECODE.COM

First printing, January 2023

Contents

List of Figures

List of Tables

Dedicated to those who appreciate LaTeX

and the work of Edward R. Tufte and Donald E. Knuth.

Preface

This documents is for the lectures of Winter school in Statistical Physics Division in Korean Physical Society.

Prerequisite

Prerequisite for "Numerical Methods for quantum systems"

Lecturer : Ji-Woo Lee

The students should prepare their laptop for the lecture.

On the laptops, the students should install some software we need.

Install Julia https://julialang.org/downloads/

It is recommended that you should install LTS (Long-term support) version.

Long-term support (LTS) release: v1.6.7 (July 19, 2022)

To learn Julia language, an online book is available.

https://benlauwens.github.io/ThinkJulia.jl/latest/book.html

Install Anaconda Anaconda is useful when we execute Julia program in the jupyter web interface.

https://www.anaconda.com/products/distribution

Depending on the system you have, you can download a proper distribution.

Check that you can run anaconda navigator and in the menu "HOME", you can launch jupyter notebook.

Connecting jupyter notebook and Julia In the Julia shell (REPL),

julia>using Pkg ; Pkg.add("IJulia")

This will enable Julia kernel in the jupyter notebook.

If you have any question, please contact jwlee@mju.ac.kr.

Part I

Exact diagonalization

Making Quantum Basis

Ising Spins

The quantum mechanics starts with the vector basis.

All the quantum operators are linear operators in $\mathbf{C}^n$

Let's start with a simple example.

Two lattice sites and for each site, the spin is located.

$$-JS_1S_2 \tag{1}$$

If $S_1(S_2)$ has two possible values $+1$ or -1, then there are 4 possible quantum states. In Dirac notation, we can write down 4 states as

$$|++\rangle, |+-\rangle, |-+\rangle, |--\rangle \tag{2}$$

The corresponding energies are

$$-J, +J, +J, -J \tag{3}$$

In summary, the Hamiltonian is diagonalized (which is boring) and the ground state is $|++\rangle$ or $|--\rangle$, which is degenerate[1].

[1] Degeneracy: two or more quantum states have the same energy eigenvalue

Boson Model

Now a second example, we consider an extended Boson Hubbard model[2]

With hard-core condition, the boson number at site i is restricted o or 1.

Then the possible quantum states for two-site bosons are

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle \tag{4}$$

[2] See *Boson localization and the superfluid-insulator transition* Matthew P. A. Fisher, Peter B. Weichman, G. Grinstein, and Daniel S. Fisher Phys. Rev. B 40, 546 – Published 1 July 1989.

The Hamiltonian is

$$H = -t(b_i^\dagger b_j + c.c.) + Vn_in_j \tag{5}$$

The diagonal elements are

$$0, 0, 0, V \tag{6}$$

Two states are connected: $|01\rangle$ and $|10\rangle$.

And we know

$$b^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle \tag{7}$$

$$b|n\rangle = \sqrt{n}|n-1\rangle \tag{8}$$

$-t$ offdiagonal term appears!

The Hamiltonian matrix becomes

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -t & 0 \\ 0 & -t & 0 & 0 \\ 0 & 0 & 0 & U \end{pmatrix} \tag{9}$$

Diagonalization

Windows

System Requirement: Windows 10, 11

1. JULIA 1.8.4 install

https://julialang.org/downloads/

Pkg.add("IJulia")

https://www.anaconda.com/products/distribution

Anaconda install

Ubuntu Linux

System Requirement: Ubuntu 22.04

1. Anaconda Install

https://linuxhint.com/install-anaconda-ubuntu-22-04/

2. Julia install

https://www.digitalocean.com/community/tutorials/how-to-install-julia-programming-language-on-ubuntu-22-04

Code: Two-site extended hardcore bosons

```
using LinearAlgebra

t=1

U=1

A = [ 0 0 0 0 ; 0 0 -t 0; 0 -t 0 0; 0 0 0 U ]

eigvals(A)

eigvecs(A)
```

Code: Hubbard model

This code works for open boudary conditions.

```
using QuantumLattices

using ExactDiagonalization

using LinearAlgebra: eigen

# define the unitcell of the square lattice

unitcell = Lattice([0.0, 0.0]; name=:Square, vectors=[[1.0, 0.0],
    [0.0, 1.0]])

# define a finite 3×4 cluster of the square lattice with open
    boundary condition

lattice = Lattice(unitcell, (3, 4))

# define the Hilbert space (single-orbital spin-1/2 complex
    fermion)

hilbert = Hilbert(site=>Fock{:f}(1, 2) for site=1:length(lattice))

# define the binary bases of the a half-filled system on the above
    cluster

bases = BinaryBases(1:12, 6) ⊗ BinaryBases(13:24, 6)

# define the terms, i.e. the nearest-neighbor hopping and the
    Hubbard interaction

t = Hopping(:t, -1.0, 1)

U = Hubbard(:U, 8.0)

# define the exact diagonalization algorithm for the Fermi Hubbard
    model

ed = ED(lattice, hilbert, (t, U), TargetSpace(bases))

# find the ground state and its energy

eigensystem = eigen(matrix(ed); nev=1)

# Ground state energy should be -4.913259209075605

print(eigensystem.values)
```

QR method

QR method is used for a dense matrix.

By iterating, it becomes more upper and upper!

Schur's decomposition theorem

$$A = Q * UQ \tag{10}$$

Lanczos method

Lanczos method is used for a sparse matrix.

Exercises

1. Write down the partition function $Z(T)$ for two-site Ising model. Make a graph as a function of temperature.
2. Develop a code for modified Lanczos method to obtain the ground state.
3. Can you generalize making a quantum states with hardcore bosons? Write down a code for it. If you include $-\mu n_i$ term, which is chemical energy term, find the quantum critical points.
4. Let's make a basis for 4-site fermion problems, which is Hubbard model.
5. With QR factorization, make a code for getting eigenvalues of a random matrix.
6. Using Lanczos algorithm, make a code for getting eigenvalues of a random matrix.

Part II

Monte Carlo simulation

Traditional Metropolis algorithm

The 10 Algorithms with the Greatest Influence on the Development and Practice of Science and Engineering in the 20th Century

At Stanford:

NEW COURSE: The Top Ten Algorithms of the Century Math 224/CS 339

We thought that, along with Scientists and Engineers, the Mathematicians, Computer Scientists and Statisticians at Stanford would benefit from a survey course covering roughly one algorithm per week. This will be a high level review including basic ideas, applications, history, pointers to available code and theory.

Persi Diaconis Gene Golub

The most important algorithms

```
{\bf Metropolis Algorithm for Monte Carlo}

Simplex Method for Linear Programming

Krylov Subspace Iteration Methods

The Decompositional Approach to Matrix Computations

The Fortran Optimizing Compiler

QR Algorithm for Computing Eigenvalues

Quicksort Algorithm for Sorting

Fast Fourier Transform

Integer Relation Detection

Fast Multipole Method
```

New Kernel adding in Julia

https://stackoverflow.com/questions/56284321/how-do-you-add-jupyter-notebook-kernels-for-prior-versions-of-julia

Metropolis algorithm

Full Diagonalization is impossible for the relatively small lattice size. For example, 4×4 lattice Hubbard model is very hard problem.

Then, what can be done?

One of things we can do is that we use a stochastic evolution method.

The Metropolis algorithm[3] is one of the Monte Calro algorithm[4].

Let's suppose the system at a state s, has a probability of $p(s)$.

Then any observable can be obtained by

$$\sum_s p(s)O(s). \tag{11}$$

If we sample correctly following $p(s)$, then this will be

$$\sum_{s_n} O(s_n) \tag{12}$$

[3] Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. (1953). *"Equation of State Calculations by Fast Computing Machines"*. Journal of Chemical Physics. 21 (6): 1087–1092. Bibcode:1953JChPh..21.1087M. doi:10.1063/1.1699114. OSTI 4390578. S2CID 1046577. There is a controversy on who first thought of this algorithm and it seems that Metropolis was just a system manager at Los Alamos National Lab.

[4] Monte Carlo algorithm was first conceived by S. Ulam. von Neumann and Metropolis joined Ulam's project on solitare.

Detailed Balance

$$p(s)p(s \to s') = p(s')p(s' \to s) \tag{13}$$

If we define $p(s \to s')$ as

$$\min(\frac{p(s')}{p(s)}, 1), \tag{14}$$

then

Suppose, $p(s') > p(s)$.

Then in Eq. (13), $LHS = p(s)$. and $p(s' \to s) = p(s)/p(s')$. So, $RHS = p(s)$.

Also suppose $p(s') < p(s)$.

Then in Eq. (13), $RHS = p(s')$. and $p(s \to s') = p(s')/p(s)$. So, $LHS = p(s)$.

Ising Model

Local update

We pick up a random site, and try to flip its spin.

See the code

For nonlocal algorithm, there are Swendsen and Wang algorithm and Wolff algorithm.

Swendsen & Wang algorithm

$$Z = \sum_s e^{-E[s]/kT} = \sum_s e^{\frac{J}{kT}\sum_{\langle n,m\rangle} S_n S_m} \tag{15}$$

$$Z = \sum_{i,jNOTm,n} e^{\frac{J}{kT}\sum_{\langle i,j\rangle} S_n S_m} \tag{16}$$

Code run

New Kernel adding in Julia

https://stackoverflow.com/questions/56284321/how-do-you-add-jupyter-notebook-kernels-for-prior-versions-of-julia

Nonlocal algorithm

Fermionic Monte Carlo method

`https://github.com/lch/dqmc_notebook`

When $U = 0$, the system becomes noninteracting fermion system. There is no correlation between spin-up and spin-down electrons.

Then we can obtain the ground state energy by filling the Fermi energy levels.

When $L \times L = 2 \times 2$, half-filled:

The g.s. energy is -8.

The energy per site is $-8/4 = -2$.

${}_4C_2 = \frac{4!}{2!2!} = 6$.

There are 36 states.

The code in ED-Hubbard works fine.

$U = 0$, $t = 1$, 2×2 lattice with periodic boundary condition and half-filled case. The g.s. energy is -8.0.

Cluster algorithm

Worm algorithm

Stochastic series expansion algorithm

Exercises

1. Check whether you can go up to 4×4 lattice size with Exact diagonalization code. Estimate the number of the possible quantum states for the half-filled case.
2. Find the energy as a function of U for two-dimensional Hubbard model with DQMC.
3. Compare the results of the exact diagonalization of 2×2 Hubbard model and those of DQMC run.

Part III

Tensor Network States

What is TNS?

A new paradigm from Quantum Information Theory.

The basic idea of tensor network states is that we decompose the many-body quantum states with the product of tensors[5].

Many body quantum state is usually writeen as

$$\Psi(i_1, i_2, \cdots, i_N) = \sum C(i_1, \cdots, i_N)|i_1, \cdots, i_N\rangle \tag{17}$$

Not all quantum states in the Hilbert space of a many-body system are equal: some are more relevant than otheres.

In particular, one can prove that low-energy eigenstates of gapped Hamiltonians with local interactions obey the so-called *area-law* for the entanglement entropy[6]. The entanglement entropy of a region of space tends to scale, for large enough regions, as the size of the boundary of the region and not as the volume.

Low-energy states of realistic Hamiltonians are not just any state in the Hilbert space: they are heavily constrained by locality so that they must obey the entanglement area-law.

Think of RG (renomalization group). RG methods keeps track of the relevant degrees of freedom to describe a system. TN states targets the most relevant corner of states.

$$\textit{Density Matrix RG} \sim \textit{TN evolution of large entanglement} \tag{18}$$

In Vidal's work, he introduced the Time-Evolulution-Block-Decimation.

Time Evolution Block Decimation

The iTEBD (infinit-TEBD) starts with two tensors,

$$|MPS\rangle = \sum \Gamma^{p_1}_{\alpha\beta}\lambda_\beta\Gamma^{p_2}_{\beta\gamma}\lambda_\gamma \tag{19}$$

Note that p_i are the physical index. For example, it can be $0, 1$ in hard-core boson case, or $\uparrow, \downarrow$ in spin case. λ's are the entanglement amplitude.

[5] *"Efficient Classical Simulation of Slightly Entangled Quantum Computations"*, Guifré Vidal, ,Phys. Rev. Lett. 91, 147902 – Published 1 October 2003.

[6] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004

The iTEBD is the same as power method.

For the time evolution, we multiply the state with an operator,

$$\exp(-\epsilon H) \tag{20}$$

where H is the Hamiltonian.

Let's think of a random state.

The random state can have the overlap with the basis

$$|R\rangle = \sum_E |E\rangle\langle E|R\rangle = |E_0\rangle\langle E_0|R\rangle + |E_1\rangle\langle E_1|R\rangle + \cdots \tag{21}$$

If we act $\exp(-\epsilon H)$ to this state,

$$\exp(-\epsilon H)|R\rangle = e^{-\epsilon E_0}|E_0\rangle\langle E_0|R\rangle + e^{-\epsilon E_1}|E_1\rangle\langle E_1|R\rangle + \cdots \tag{22}$$

$$e^{-\epsilon E_0}(|E_0\rangle\langle E_0|R\rangle + e^{-\epsilon(E_1-E_0)}|E_1\rangle\langle E_1|R\rangle + \cdots \tag{23}$$

As we multiply time-evolution operator, the amplitude of excited states are diminishing.

The same thing happens in TEBD.

If ϵ is reasonably small,

$$\exp(-\epsilon H) \sim \exp(-\epsilon H_{12})\exp(-\epsilon H_{21}) \tag{24}$$

Let's focuss on H_{12}.

$$\Theta(p_1', p_2') = \langle p_1' p_2'| \exp(-\epsilon H)|p_1, p_2\rangle \Gamma^{p_1}\lambda_1\Gamma^{p_2} \tag{25}$$

This new tensor is decomposed to a new Γ's and λ's.

The mathematical formalism for the decomposion is "*singular value decomposition*".

We iterate this procedure until we get the reliable values for the ground state.

Density matrix renormalization group

S. White's seminal paper.

Numerical RG with density matrix.

Usually, numerical RG was done by taking the low-energy states.

In DMRG, the states with maximized density matrix eigenvalues are kept.

DMRG with MPO

MPO: Matrix product operators.

For the Transverse Field Ising model[7],

[7] Subir Sachdev. Quantum Phase Transitions. Cambridge University Press, 2011.

$$H = -J \sum_i (\sigma_i^z \sigma_{i+1}^z + g\sigma_i^x) \tag{26}$$

With Bogoliubov transformation, $e_k = -2J(1 + g^2 - 2gcosk)^{1/2}$.

$$E = \frac{1}{2\pi} \int_0^\pi e_k dk \tag{27}$$

MPO

$$H = \begin{pmatrix} I & 0 & 0 \\ \sigma^x & 0 & 0 \\ \sigma^z & \sigma^x & 0 \end{pmatrix} \tag{28}$$

Consider $H_1 \otimes H_2$.

$$\begin{pmatrix} I_1 & 0 & 0 \\ \sigma_1^z & 0 & 0 \\ \sigma_1^x & \sigma_1^z & I_1 \end{pmatrix} \otimes \begin{pmatrix} I_2 & 0 & 0 \\ \sigma_2^z & 0 & 0 \\ \sigma_2^x & \sigma_2^z & I_1 \end{pmatrix} = \begin{pmatrix} I_1 \otimes I_2 & 0 & 0 \\ \sigma_1^z \otimes I_2 & 0 & 0 \\ \sigma_1^x \otimes I_2 + \sigma_1^z \otimes \sigma_2^z + I_1 \otimes \sigma_2^x & I_1 \otimes \sigma_2^z & I_1 \otimes I_2 \end{pmatrix} \tag{29}$$

At the right boundary, $\times(I, 0, 0)^T$, then It becomes $(I_1 \otimes I_2,\ \sigma_1^z \otimes I_2,\ \sigma_1^x \otimes I_2 + \sigma_1^z \otimes \sigma_2^z + I_1 \otimes \sigma_2^x)$.

At the left boundary, $(0, 0, I)\times$, then it becomes

$$\sigma_1^x \otimes I_2 + \sigma_1^z \otimes \sigma_2^z + I_1 \otimes \sigma_2^x \tag{30}$$

which is *really* two-site Hamiltonian.

Exercises

1. For a hard-core boson system, find the operator $\exp^{-\epsilon H_{12}}$ in a closed form.
2. Develop a code for the original S. White's DMRG algorithm.
3. How can you make finite-size DMRG algorithm?
4. Develop a code for TFIM model by modifying the XX code explained in the lecture.

Term Project

1. (Challenging, TNS) Find some references on the solution of two-dimensional quantum problems with tensor network states and discuss the way they achieved.
2. (Challenging, Monte Carlo) Compare the Monte Carlo simulations using Metropolis, Swendsen, and Wolff algorithms.

Bibliography

[1] T. Xiang, J. Lou, Z. Su, Phys. Rev. B 64, 104414 (2001); O. Legaza, J. S olyom, cond-mat/0305336; O. Legaza, J. S olyom, Phys. Rev. B 70, 205118 (2004); G. Vidal, J. I. Latorre, E. Rico, A. Kitaev, Phys. Rev. Lett. 90 227902 (2003); P. Calabrese, J. Cardy, JSTAT 0406:P06002 (2004); M. Srednicki, Phys. Rev. Lett. 71, 666 (1993); M. Plenio, J. Eisert, J. Dreißig, M. Cramer, Phys. Rev. Lett. 94, 060503 (2005)

Index