Entanglement in Topological Phases

Dylan Liu

August 31, 2012

Abstract

In this report, the research conducted on entanglement in topological phases is detailed and summarized. This includes background developed and an explanation of the structure of the code written it MATLAB to compute various aspects of entanglement for topological systems. Topological phases are generally those that cannot be described by Landau symmetry-breaking, including the fractional quantum Hall states (fQHE). From the reading, it is found that the entanglement entropy (EE) and entanglement spectrum (ES) are useful tools to characterize topological phases. Further, the code written is successful in reproducing the results of several papers that calculate the EE and ES.

1 Introduction

The organization of this report is as follows: first an overview of topological phases, then a brief summary of important and relevant aspects of entanglement, lastly followed by examples and recent developments in addition to my work so far.

2 Topological Phases

Examples include fractional quantum Hall states, chiral spin states, and depending on precise definitions, topological insulators, etc. Fractional quantum Hall states are the only experimentally realized examples. They consist of a Gallium Arsenide heterostructure. They cannot be characterized by Landau symmetry-breaking. That is, the normal breaking of symmetries in phases which distinguishes them from each other is not applicable to these states of matter. Instead, they are characterized by other properties, such as non-Abelian geometric phase, fractional statistics and fractional charge, and topological entanglement entropy. Though most of the observed phases are Abelian, there are suspected to be non-Abelian quasiparticles. The name comes from the effective field theories that describe the low-energy chiral spin states. These theories are actually topological quantum field theories, which means they are invariant of the spacetime metric. Topological phases are interesting and relevant because they are an indication that Landau symmetry-breaking is incomplete. Previously, this symmetry breaking description was thought to encompass all continuous phase transitions and orders of matter. Another source of potential for these phases of matter is topological quantum computing. The idea is to employ a quasiparticle or excitation of a topological phase that is non-Abelian and has anyonic statistics. Braids are then made in spacetime with the worldlines of these particles. Different braids represent different logical operations and the advantage over regular quantum computing is that topological quantum computing is resistant to local perturbations. That is, only the topology of the braids matters for doing computations and calculations.

3 Entanglement

The idea of entanglement was first discussed by Einstein, Podolsky, and Rosen in the mid 1930s. However, the word entanglement was not used until Schrodinger later on. They presented it very differently than it is explained today, but the idea is the same. In a quantum state in which there is a superposition of states of multiple particles which are separated, there are correlations in the system that are non-local. Einstein, Podolsky, and Rosen initially concluded that this meant quantum mechanics was incomplete, and that some “hidden variable” existed that described the physics in a local way. Later on, John Bell proved that this is incorrect, and that no local, hidden variable theory can describe the physics of quantum mechanics.

3.1 Schmidt decomposition

Schmidt decomposition is critical for investigating the entanglement of a system. One generally observes the entanglement between subparts of the whole system. In order to compute this entanglement, one has to write the system in terms of tensor products of states from each part and Schmidt decomposition allows for this. Schmidt decomposition involves taking a partition of a Hilbert space, $\mathcal{H}$ into to parts, $\mathcal{H}_A$ and $\mathcal{H}_B$. Recall singular value decomposition: take an $M \times N$ matrix (real or complex) and it can be written as $U\Sigma V^*$, where U is an $M \times M$ unitary matrix, Σ is a $M \times N$ rectangular, diagonal, non-negative real matrix, and V is a $N \times N$ unitary matrix. We can (by singular value decomposition applied to a matrix representation of the space) write any element of $\mathcal{H}$ as $|\psi\rangle = \sum_{i=1}^{n} \rho_i |u_i\rangle \otimes |v_i\rangle$, where n is $\min(\dim \mathcal{H}_A, \dim \mathcal{H}_B)$ and u_i, v_i form orthogonal bases for $\mathcal{H}_A, \mathcal{H}_B$ and ρ_i non-negative constants. This is relevant for entanglement in topological phases because we will compute the entanglement between the two partitions of the space. The partitions can be in real-space, orbital-space, particle-space, and interpolations between them as well.

3.2 Entanglement Entropy

The entanglement entropy is the most primitive tool with which we can examine the degree of entanglement in a system. It is based on the von Neumann entropy of quantum mechanics, which is further rooted in the Gibbs entropy from thermodynamics. The idea is to compute take a state $|\psi\rangle$, and consider the density matrix, $\rho = |\psi\rangle \langle\psi|$. Then calculate the reduced density matrix for one of the partitions, either A or B, the result $S_A = S_B$. The von Neumann entropy is then $-\mathrm{Tr}\,(\rho_A \ln \rho_A)$. The functional form is very clearly similar to the analogous term in thermodynamics. For topological phases, a scaling law has been seen to emerge in the computation of this quantity. The scaling goes as $S_A = \alpha L - \gamma + \mathcal{O}\left(\frac{1}{L}\right)$, where L is the distance scale describing the physical boundary of the partition. There is a term that goes as the boundary, terms that go to 0 as the system becomes large (the thermodynamic limit), and lastly, a term that is specifically topological, γ. This is the topological entanglement entropy and characterizes some aspects of topological phases.

3.3 Entanglement Spectrum

The next most sophisticated tool for examining entanglement in topological phases is the entanglement spectrum. We can write ρ_A from a Schmidt decomposition as $\exp\left(-\frac{\xi_i}{2}\right)$ and $\{\xi_i\}$ is the entanglement spectrum. The idea is to define a pseudo-Hamiltonian, $\mathcal{H}_A \equiv -\log \rho_A$, and then the eigenvalues of this operator are pseudo-energy eigenvalues. When calculating the entanglement spectrum, there are a few different ways to partition $\mathcal{H}$. We have the orbital cut (OP) which partitions the space based on momentum, we have the real space cut (RS) which uses position space to partition, we can also partition the Hilbert space by simply labeling the particles of the system and splitting them. The entanglement spectrum has been seen to reveal more about topological phases than the entanglement entropy alone. Li and Haldane asserted that it is natural to consider this pseudo-Hamiltonian as that of a CFT describing the edge physics of the system. They asserted that the multiplicities of the entanglement spectrum would match the multiplicities of the energy spectrum of the CFT for the edge.

4 Recent Developments

In the past several years, since the development of the entanglement entropy and spectrum to investigate entanglement in topological phases, research has tried to use the entanglement spectrum to make predictions about the edge physics of these states. Here we present some of the most recent work, particularly the use of a real-space partition in computing the entanglement spectrum.

4.1 Real-space entanglement spectrum of QH systems (Dubail, Read, Rezayi)

They define the entanglement spectrum in terms of the Schmidt decomposition and discuss briefly the different possibilities for partitions. DRR claim that in the real space partition there is locality along the cut, i.e. the correlations are short-ranged near the partition. They also demonstrate that this is not the case in OP, where the correlations are long-ranged along the cut.

Next, the Schmidt decomposition on the sphere is set up. Then, DRR show that for the $\nu = 1$ QH state, the ES approaches the edge state spectrum, however the multiplicities do not match (of the ES and the edge state).

Next, DRR conjecture an asymptotic scaling property: as the number of particles in the system$\rightarrow \infty$, the set of $\Delta\xi_i = \xi_i - \xi_0$ approach the energy levels of a Hamiltonian that is the integral of a sum of local operators in a $1+1$ dimensional field theory on a circle for all $\Delta N_A, \Delta L_z^A$, where the following are defined: $\Delta N_A \equiv N_A - N_{A0}$, $\Delta L_z^A = L_z^A - L_{z0}^A$. They note that in the simplest case the low-lying spectrum should be a straight line.

5 Orbital partition for $\frac{1}{3}$ Laughlin state

The simplest examples in which to examine the topological entanglement are those of the fractional quantum Hall states, specifically, the fermionic Laughlin states. For a filling fraction of $\nu = \frac{1}{m}$, the Laughlin state for a single particle is $\frac{z^m}{\sqrt{2\pi 2^m m!}} \exp\left(-\frac{1}{4}\left|z\right|^2\right)$. For N particles, we have $\propto \prod_{i<j} \left(z_i - z_j\right)^m e^{-\frac{1}{4}\sum_i |z_i|^2}$. Consider first, $N = 2$ on a spherical surface rather than a plane. The wavefunction can be written as $\left(a_1^\dagger b_2^\dagger - a_2^\dagger b_1^\dagger\right)^3 = \left[\left(a_1^\dagger\right)^3\left(b_2^\dagger\right)^3 - \left(a_2^\dagger\right)^3\left(b_1^\dagger\right)^3\right] + 3\left[\left(a_1^\dagger\right)\left(a_2^\dagger\right)^2\left(b_1^\dagger\right)^2\left(b_2^\dagger\right) - \left(a_2^\dagger\right)\left(a_1^\dagger\right)^2\left(b_2^\dagger\right)^2\left(b_1^\dagger\right)\right]$. In terms of particle notation, this is equivalent to $\left|-3,3\right\rangle + 3\left|-1,1\right\rangle$ up to factors from the Schwinger boson operators. In this case the factors are such that the coefficients in front of each term are equivalent. So we can write the density matrix as a diagonal 2×2 with entries $\frac{1}{2}$.

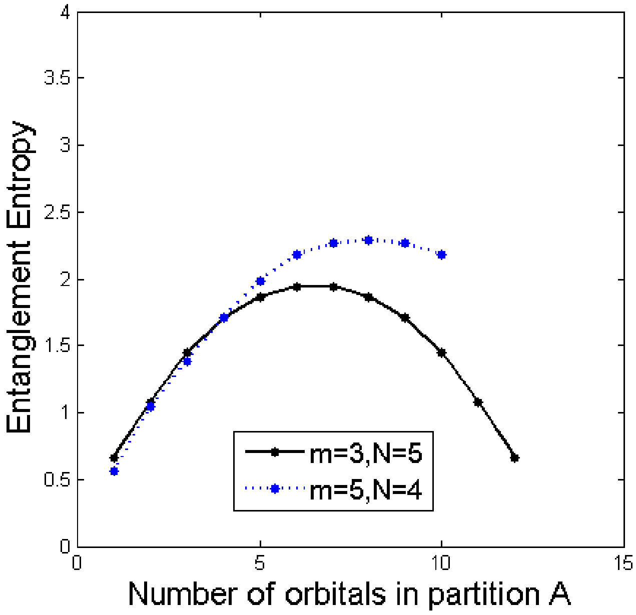


Fig. 1: Entanglement entropy for various partitions of Laughlin states for $m = 3, 5$ and $N = 5, 4$

6 In Progress

6.1 Numerics

I have written code in MATLAB that is able to compute the entanglement spectra and entropy for various topological states of matter. The code is included as an appendix to this report. The code is written as several functions and can currently compute the entropy and spectra for Laughlin states $\frac{1}{m}$ on the plane and on the sphere. The code takes as inputs for the Laughlin states the value of m and the number of particles, N. Currently the code can implement an orbital partition. The analytic forms of the states are computed and then reduced from the naturally anti-symmetrized form to accomodate the particle notation. Then it is converted to an orbital notation with 1 or 0 whether the orbital is occupied or not. Next the reduced density matrix is computed depending on the partition specified. For sufficiently high m and N, there are non-diagonal terms in the reduced density matrix, so it must be diagonalized. Presently the code implements the native MATLAB diagonalization, but this may be modified in the future for efficiency.

Plots have been made to compare the results of the code to previous work by Haque, et al. They show that the code is likely just as robust when it comes to computing these entropies and spectra. However, for more than $N = 5$ particles, there is significant computing power necessary to determine the Laughlin states, so a more abstract method using characters of matrix representations of the symmetric group are employed. This code does not implement that yet. The next steps include converting the code from MATLAB to a language like C++ or Fortran. Furthermore, the code will be adapted to accommodate more than the Laughlin states and for computing systems with more than $N = 5$ particles.

7 References

1. H. Li and F. D. M. Haldane, Phys. Rev. Lett. 101, 010504 (2008), arXiv:0805.0332.

2. A. Kitaev and J. Preskill, Phys. Rev. Lett. 96, 110404 (2006), hep-th/0510092.

3. M. Levin and X.-G. Wen, Phys. Rev. Lett. 96, 110405 (2006), cond-mat/0510613.

4. R. B. Laughlin, Phys. Rev. Lett. 50, 1395 (1983).

5. I. D. Rodrguez and G. Sierra, Phys. Rev. B 80, 153303 (2009), arXiv:0811.2188.

6. J. Dubail, N. Read, and E. Rezayi, arXiv:1111.2811.

7. Nayak, C. et al, 2008, “Non-Abelian anyons and topological quantum computation,” Rev. Mod. Phys. 80, 1083-1151.

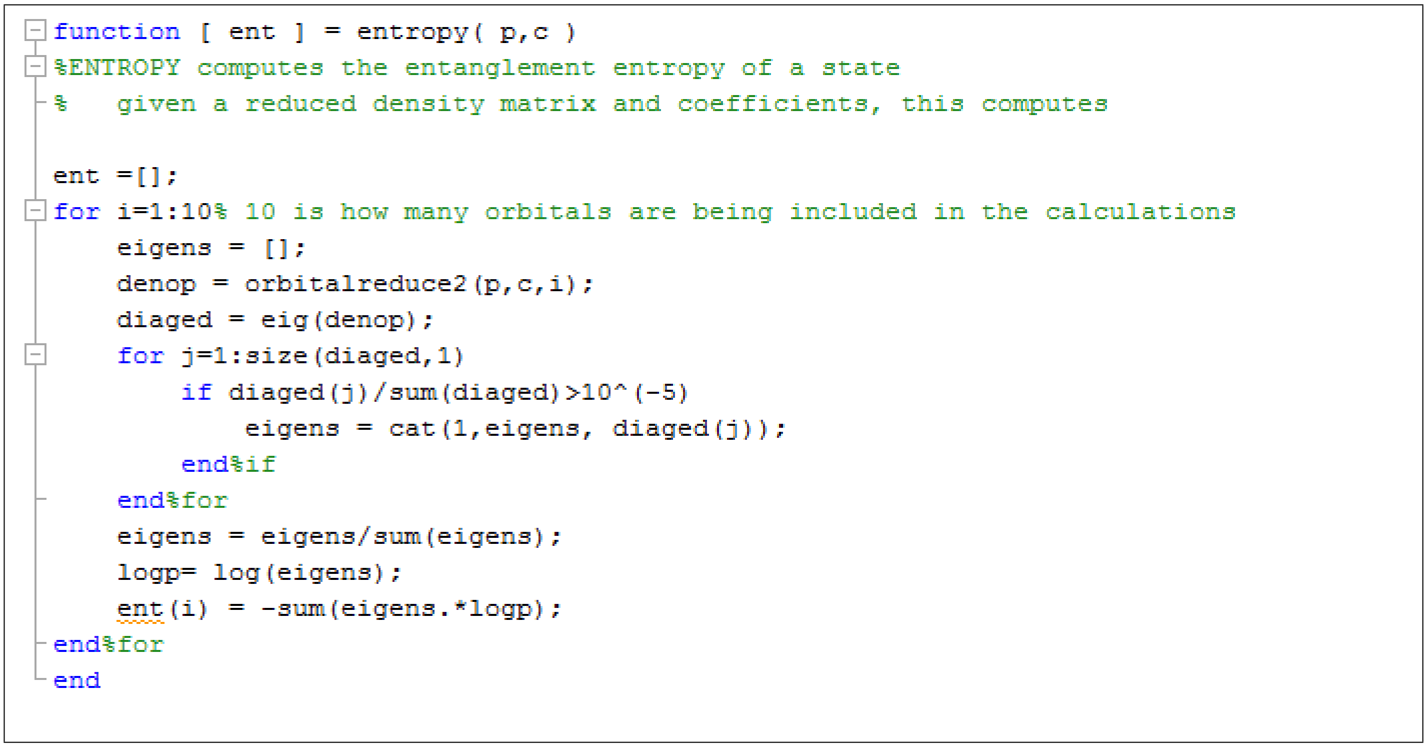
8. Einstein, A., B. Podolsky, N. Rosen, 1935, "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?" Phys. Rev. 47 (10), 777–780.

9. Sterdyniak, A., et al., 2012, "Real-space entanglement spectrum of quantum Hall states" Phys. Rev. B 85, 125308.

10. Haque, M., Zozulya, O., Schoutens, K., 2007, Phys. Rev. Lett. 98, 060401

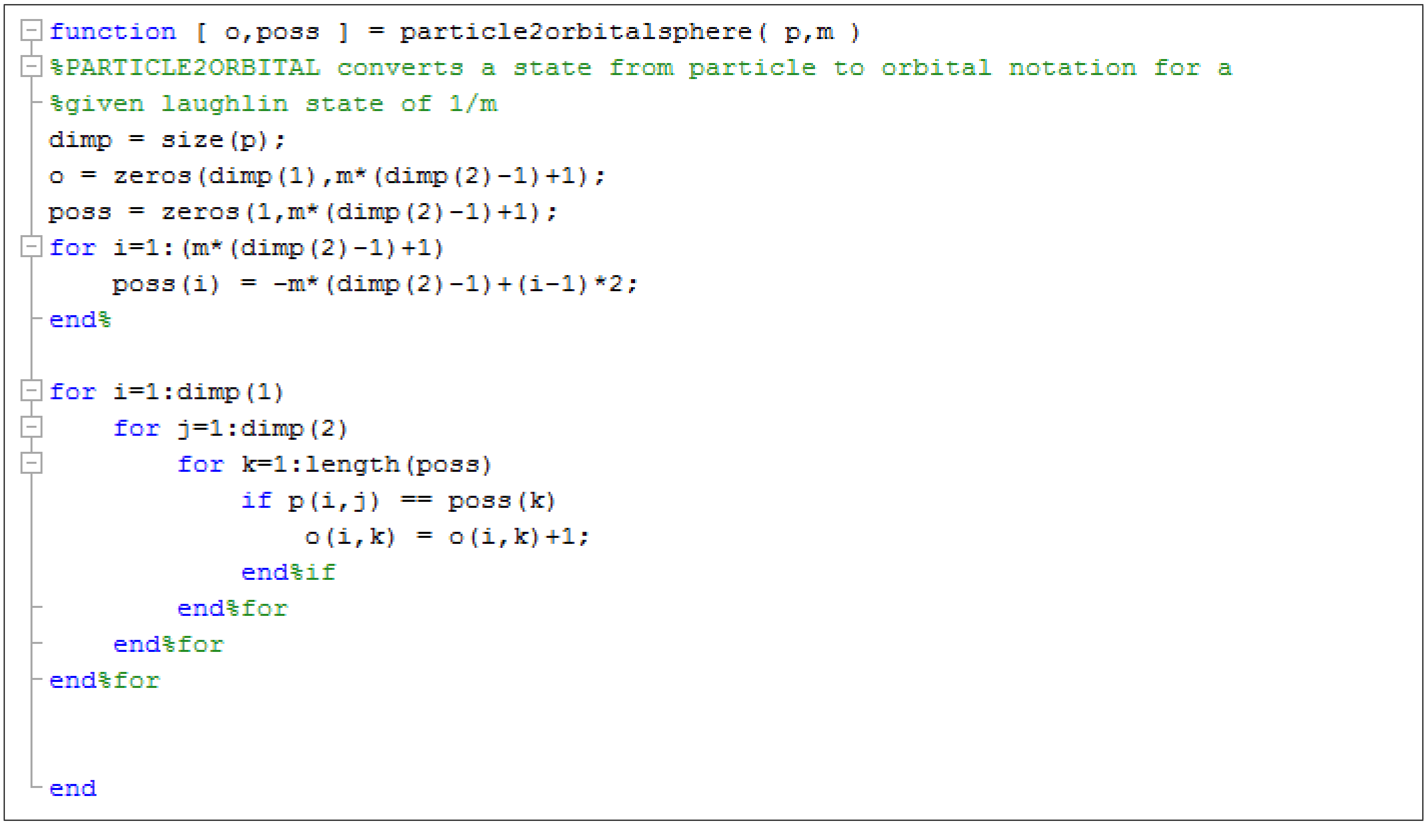
8 Appendix: MATLAB code

8.1 Entanglement Entropy calculation



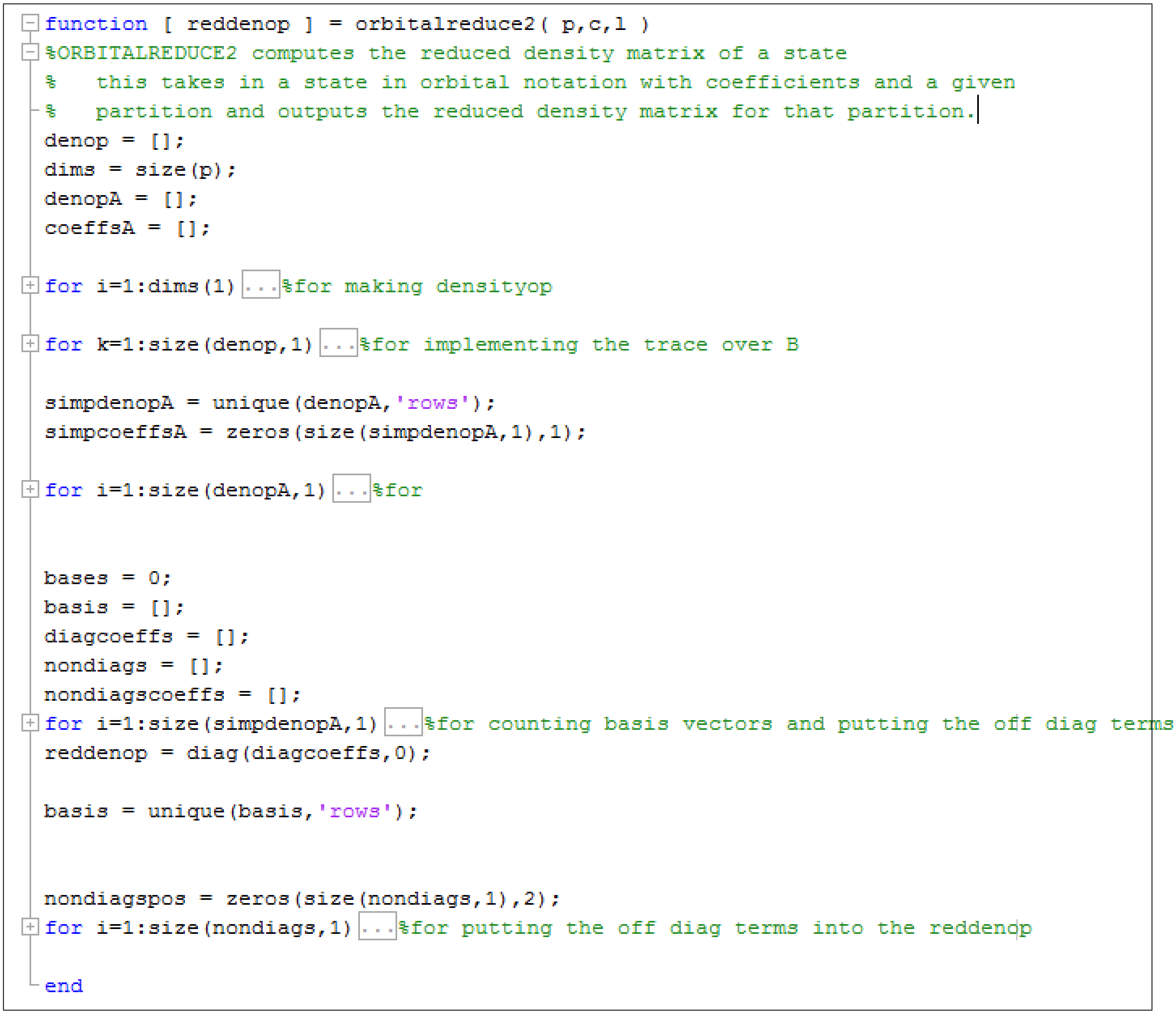
```
function [ ent ] = entropy( p,c )
%ENTROPY computes the entanglement entropy of a state
%   given a reduced density matrix and coefficients, this computes

ent =[];
for i=1:10% 10 is how many orbitals are being included in the calculations
    eigens = [];
    denop = orbitalreduce2(p,c,i);
    diaged = eig(denop);
    for j=1:size(diaged,1)
        if diaged(j)/sum(diaged)>10^(-5)
            eigens = cat(1,eigens, diaged(j));
        end%if
    end%for
    eigens = eigens/sum(eigens);
    logp= log(eigens);
    ent(i) = -sum(eigens.*logp);
end%for
end
```

8.2 Convert from particle to orbital notation



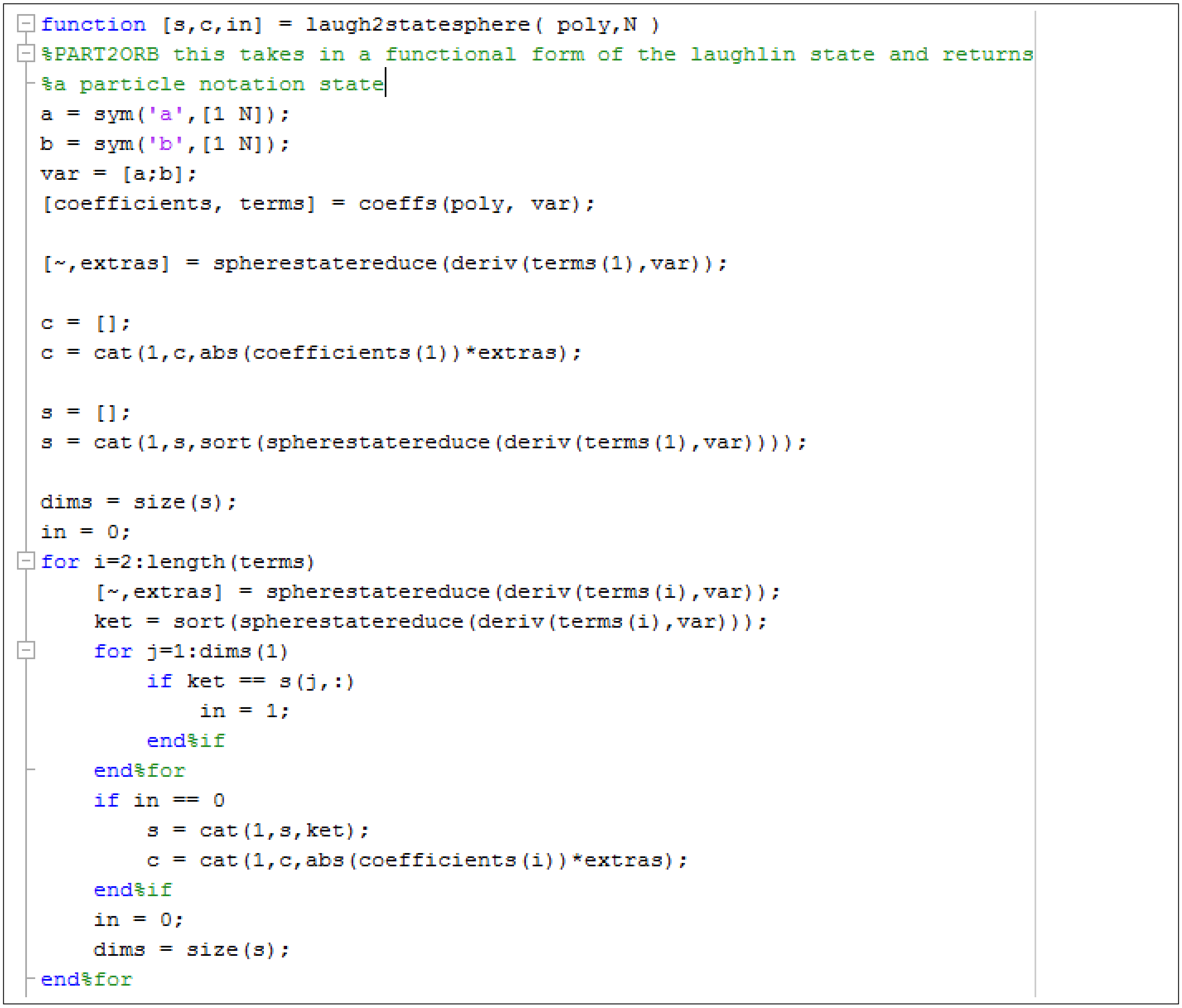
```
function [ o,poss ] = particle2orbitalsphere( p,m )
%PARTICLE2ORBITAL converts a state from particle to orbital notation for a
%given laughlin state of 1/m
dimp = size(p);
o = zeros(dimp(1),m*(dimp(2)-1)+1);
poss = zeros(1,m*(dimp(2)-1)+1);
for i=1:(m*(dimp(2)-1)+1)
    poss(i) = -m*(dimp(2)-1)+(i-1)*2;
end%

for i=1:dimp(1)
    for j=1:dimp(2)
        for k=1:length(poss)
            if p(i,j) == poss(k)
                o(i,k) = o(i,k)+1;
            end%if
        end%for
    end%for
end%for


end
```

8.3 Reduced density matrix



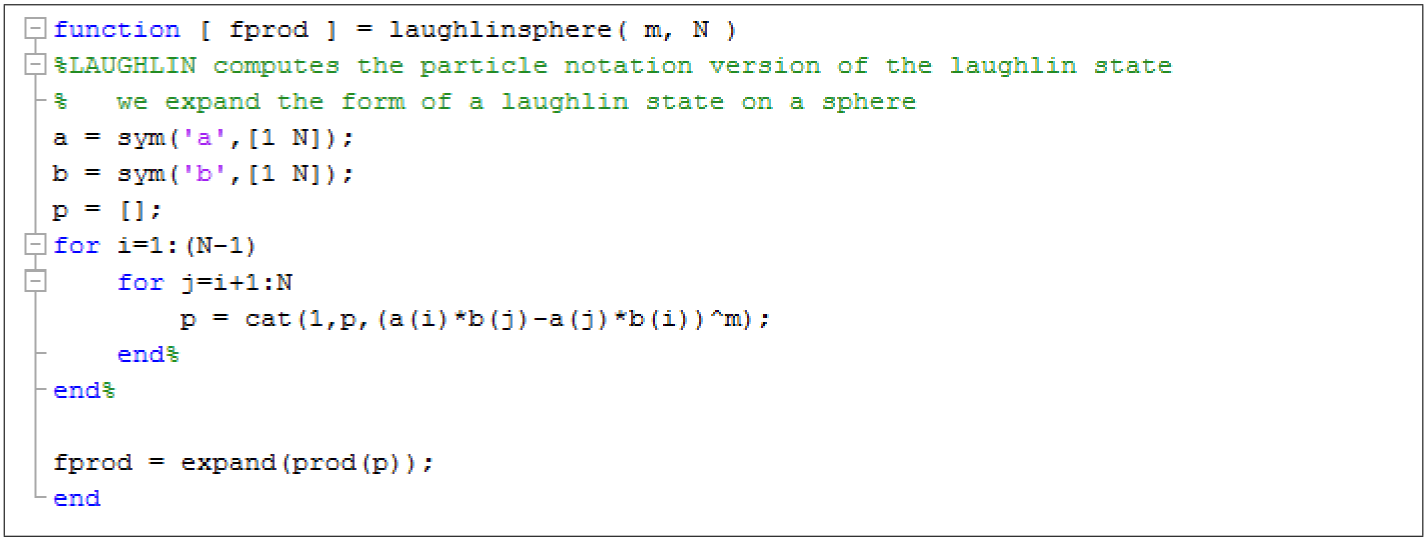
```
function [ reddenop ] = orbitalreduce2( p,c,l )
%ORBITALREDUCE2 computes the reduced density matrix of a state
%   this takes in a state in orbital notation with coefficients and a given
%   partition and outputs the reduced density matrix for that partition.
denop = [];
dims = size(p);
denopA = [];
coeffsA = [];

for i=1:dims(1) ...%for making densityop

for k=1:size(denop,1) ...%for implementing the trace over B

simpdenopA = unique(denopA,'rows');
simpcoeffsA = zeros(size(simpdenopA,1),1);

for i=1:size(denopA,1) ...%for


bases = 0;
basis = [];
diagcoeffs = [];
nondiags = [];
nondiagscoeffs = [];
for i=1:size(simpdenopA,1) ...%for counting basis vectors and putting the off diag terms toge
reddenop = diag(diagcoeffs,0);

basis = unique(basis,'rows');


nondiagspos = zeros(size(nondiags,1),2);
for i=1:size(nondiags,1) ...%for putting the off diag terms into the reddenop

end
```

8.4 Functional form to state notation



```
function [s,c,in] = laugh2statesphere( poly,N )
%PART2ORB this takes in a functional form of the laughlin state and returns
%a particle notation state
a = sym('a',[1 N]);
b = sym('b',[1 N]);
var = [a;b];
[coefficients, terms] = coeffs(poly, var);

[~,extras] = spherestatereduce(deriv(terms(1),var));

c = [];
c = cat(1,c,abs(coefficients(1))*extras);

s = [];
s = cat(1,s,sort(spherestatereduce(deriv(terms(1),var))));

dims = size(s);
in = 0;
for i=2:length(terms)
    [~,extras] = spherestatereduce(deriv(terms(i),var));
    ket = sort(spherestatereduce(deriv(terms(i),var)));
    for j=1:dims(1)
        if ket == s(j,:)
            in = 1;
        end%if
    end%for
    if in == 0
        s = cat(1,s,ket);
        c = cat(1,c,abs(coefficients(i))*extras);
    end%if
    in = 0;
    dims = size(s);
end%for
```

8.5 Compute polynomial expansion of Laughlin state



```
function [ fprod ] = laughlinsphere( m, N )
%LAUGHLIN computes the particle notation version of the laughlin state
%   we expand the form of a laughlin state on a sphere
a = sym('a',[1 N]);
b = sym('b',[1 N]);
p = [];
for i=1:(N-1)
    for j=i+1:N
        p = cat(1,p,(a(i)*b(j)-a(j)*b(i))^m);
    end%
end%

fprod = expand(prod(p));
end
```