Document

Shi Feng

1 Entanglement Entropy

In this section we describe in detail how Entanglement-related properties are to be calculated. First of all let's see how joint operators act on wavefunctions. Let $|\Psi\rangle$ be an arbitrary pure state from ED, which can be factorized into a bipartite wavefunction:

$$|\psi\rangle = \sum_{l,r} \omega_{lr} |\psi_l\rangle |\psi_r\rangle \equiv \hat{\Omega} \tag{1.1}$$

where we defined $\hat{\Omega}$ to be the matrix written under the bipartite basis. The right reduced density matrix (RDM) is then obtained by tracing out the left dofs:

$$\begin{aligned} \rho_r = Tr_l[|\psi\rangle\langle\psi|] &= \sum_{l''} \sum_{l,r} \sum_{l',r'} \omega_{lr} \omega^*_{r'l'} \langle\psi_{l''}|\psi_l\rangle |\psi_r\rangle \langle\psi'_r| \langle\psi'_l|\psi_{l''}\rangle \\ &= \sum_{l''} \sum_{l,r} \sum_{r',l'} \omega_{lr} \omega^*_{l'r'} \delta_{l,l''} |\psi_r\rangle \langle\psi'_r| \delta_{l',l''} \\ &= \sum_{lr,r'} \omega_{lr} \omega^*_{r'l} |\psi_r\rangle \langle\psi'_r| \end{aligned} \tag{1.2}$$

we can rewrite this as:

$$\rho_r = \sum_{lr,r'} \omega^*_{r'l} \omega_{lr} |\psi_r\rangle \langle\psi'_r| = \sum_{r,r'} \left(\sum_l \omega^*_{r'l} \omega_{lr} \right) |\psi_r\rangle \langle\psi'_r| \tag{1.3}$$

that is, in the basis of $|\psi_r\rangle \langle\psi'_r|$ the elements of RDM are

$$\rho_{r',r} = \sum_l \omega^*_{r'l} \omega_{lr} = \hat{\Omega}^\dagger \hat{\Omega} \tag{1.4}$$

In the same way we can show the left RDM by tracing out right is simply:

$$\rho_{l',l} = \hat{\Omega} \hat{\Omega}^\dagger \tag{1.5}$$

In other words, if we can construct the bipartite wavefunction explicitly as a matrix form, the RDM is simply a matrix product of it with its complex conjugation. To do this, note that the tensor product of the left and right (or sys and evn) follows a "distributive" pattern. A trivial example to make this clear is as follows:

$$v \otimes w = \begin{pmatrix} v1 \\ v2 \end{pmatrix} \otimes \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} v_1 w_1 \\ v_1 w_2 \\ v_1 w_3 \\ v_2 w_1 \\ v_2 w_2 \\ v_2 w_3 \end{pmatrix} \simeq \begin{pmatrix} v_1 w_1 & v_1 w_2 & v_1 w_3 \\ v_2 w_1 & v_2 w_2 & v_3 w_3 \end{pmatrix} = \hat{\Omega} \tag{1.6}$$

where v and w represents the left and right block of the pure state. That is, the factorization of left and right block is simply a reshaping process of a $dim(\psi) = dim(v) \times dim(w)$ vector into a matrix with $dim(v)$ rows and $dim(w)$ columns, which is can be easily achieved by the numpy command:

```
Omega = numpy.reshape(wavefunction, (left.size, right.size))
```

which gives the $\hat{\Omega}$ we want. Hence the right RDM is

```
right_RDM = numpy.dot(np.transpose(Omega.conj()), Omega)
```

and the left RDM is

```
left_RDM = numpy.dot(Omega, numpy.transpose(Omega.conj()))
```

The procedure descibed above is implemented in the *pwavefunction* class (p for partition) defined in *src/Wavefunction.py*.