Preliminary: loading routines and an example MERA

For these problems we provide a Mathematica code to work with tensor networks, initialising a MERA network and finding the ground state. Also included are some sample MERAs, notably for the Ising and the Heisenberg model with bond dimension $\chi = 6$. To get the package please download it from sites.google.com/site/wilkevanderschee/MERA.

To use the package one typically creates an empty notebook in the same directory of the package and loads the package with the following two statements:

```
SetDirectory[NotebookDirectory[]];
<< MERA`
```

The function `loadMERA["ising6.dat"]` will then load the pre-computed MERA for the critical Ising model with bond dimension 6, with the following Hamiltonian:[1]

$$H_{\text{Ising}} = -\sum_r \left(\sigma_z^{[r]} + \sigma_x^{[r]} \sigma_x^{[r+1]} \right), \tag{1}$$

This in particular loads the variables `w`, `u`, `oper` and `dens`, which are the isometries, unitaries, the (effective) two-site Hamiltonian and (effective) two-site density matrices at each layer of the MERA. Note that the density matrices represent the entire network, when tracing out the part outside the causal cone of the density matrix in question.

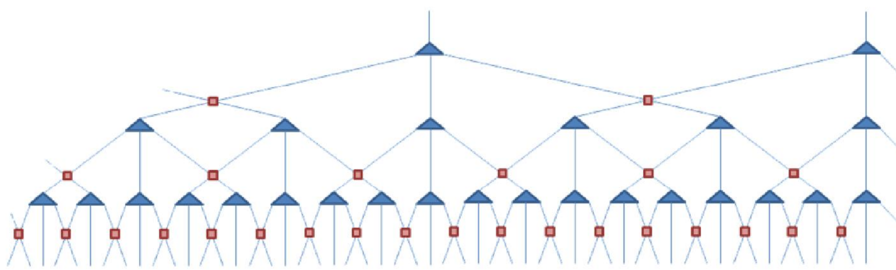


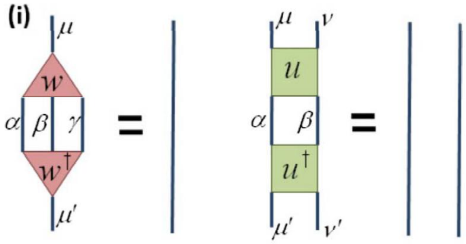
Figure 1: The tensor structure of MERA, in principle continuing indefinitely for an infinite scale invariant MERA. Typical MERAs are translationally invariant, and contain a few (5 or so) non-identical layers (after which all layers are assumed to be equal by scale invariance) and hence is specified by about 5 isometries and 5 unitaries. The MERA package however also handles non-translationally invariant MERAs for finite (periodic) systems, in which case all tensors are different, and in which case the numbering of the tensors is defined as in this figure.

1 Tensor contractions (easy/medium)

1. Verify that `w` and `u` contain a two-dimensional list of rank 4 tensors, the first index going from 1 till Nlayers, and the second index having length 1 for a translationally invariant MERA (as is the case here). What are the dimensions of the different legs, and do you understand why? (hint: use the command `Dimensions`)

2. Verify that `w` and `u` are actual isometries and unitaries; first do this using *Mathematica*'s `Sum` command. You will need that the first index of an isometry is the upper leg/index and the other three are the lower legs/indices (i.e. order $\mu\alpha\beta\gamma$ in figure 2), for the unitaries the legs are numbered $\mu\nu\alpha\beta$, according to the same figure.

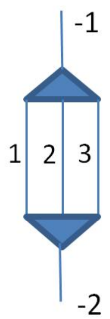
[1] The package closely follows 0707.1454 (by Glen Evenbly and Guifre Vidal), and it is also easy to compute an optimised MERA (the Ising model presented is produced in 30 minutes by `domodel[initIsing, 6]` with `initIsing` defining the Hamiltonian)



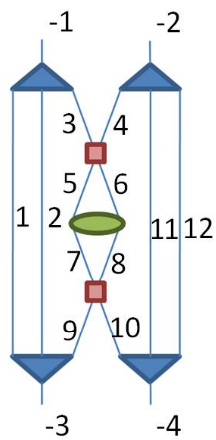
(ii)
$$\sum_{\alpha,\beta,\gamma}\left(w\right)^{\mu}_{\alpha\beta\gamma}\left(w^{\dagger}\right)^{\alpha\beta\gamma}_{\mu'}=\delta_{\mu\mu'}$$
$$\sum_{\alpha,\beta}\left(u\right)^{\mu\nu}_{\alpha\beta}\left(u^{\dagger}\right)^{\alpha\beta}_{\mu'\nu'}=\delta_{\mu\mu'}\delta_{\nu\nu'}$$

Figure 2: The isometric and unitary identities for `w` and `u` represented as tensor networks (left) or as mathematical identities (right) (figure form 0707.1454).

3. Next, we will verify the same through the `ncon` function (original from 1402.0939), which is used for all tensor network contractions[2]. It takes as arguments a list of tensors, a list of leg numbers for all these tensors and optionally the order in which the legs should be contracted (sequentially if omitted). In figure 3 two examples are presented. What properties do these list of tensors and legs need to satisfy in order to define a valid tensor contraction? Looking at `nconc`, which checks if the `ncon` input is valid; does it include all your conditions?




```
ncon[{w,w*},
{{-1,1,2,3},
 {-2,1,2,3}}]
```




```
ncon[{w,w*,w,w*,u,u*,o},
{{-1,1,2,3},
 {-3,1,2,9},
 {-2,4,11,12},
 {-4,10,11,12},
 {3,4,5,6},
 {7,8,9,10},
 {5,6,7,8}},
??
]
```

Figure 3: Two example tensor networks, with their corresponding input in `ncon`. Upper triangles are isometries `w`, whereas downward triangles are the hermitian conjugate. For unitaries `u` the hermitian conjugation is always implied by the network. The green tensor in the right network could be an operator, with the network being an ascending superoperator (as in 0707.1454). As can be seen we followed the order of the indices of the tensors as in figure 2. Note also that the order of the 'open legs' is specified sequentially with negative numbers.

4. Lastly, compute the expectation function of the spin in each direction, as well as the Hamiltonian. This can be done using the effective density matrix `dens` on the lattice layer (see lecture). The analytic ground state energy density for the Hamiltonian (1) equals $-4/\pi$; how well does this MERA approximate the ground state energy?

5. (optional) What would be the optimal way to contract the right tensor network in figure 3? How many multiplications have to be performed? Would this change if the operator is shifted one place to the left or right? (answers can be checked using the routine `findsequence[legs]`)

[2] While not needed for the problem set, it may be worthwhile to have a look at the interpretation of the `ncon` function. In particular, it cleverly uses `Transpose`, such that a single tensor contraction is just a dot product.