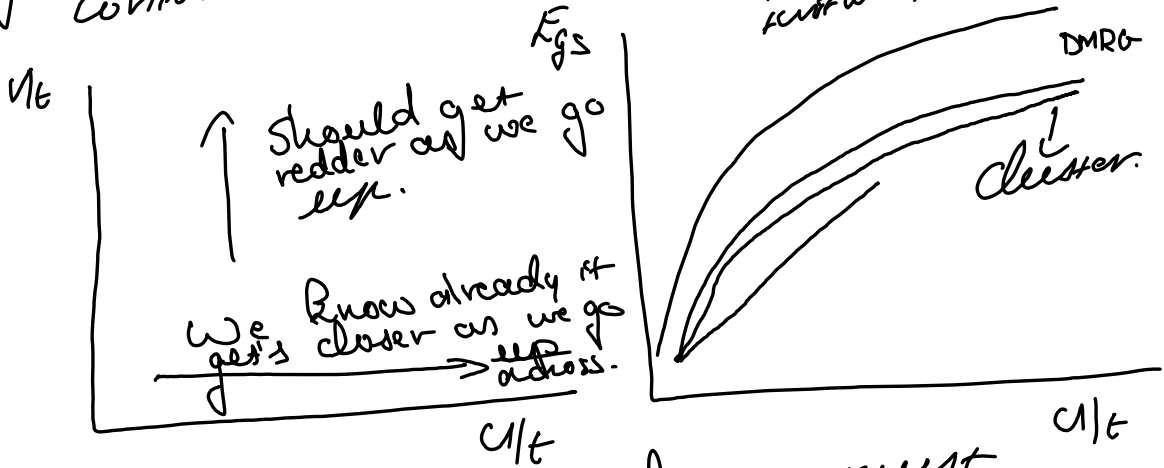


\$ 06/07/2015

OK!

Let's try to ~~not~~ write code for AAH.  
 The goal of today is to have a graph of the agreement between DMRG and the matched case, as a function of  $U/t$  and  $V/t$ .

The natural thing to do is a heatmap, but I think we'd probably get a lot more insight from just contour curves. We can do both:



I think the main prediction must be that we get closer in ~~energy~~ relative energy as  $V/t \rightarrow \infty$ . There must be some signal as we go up.

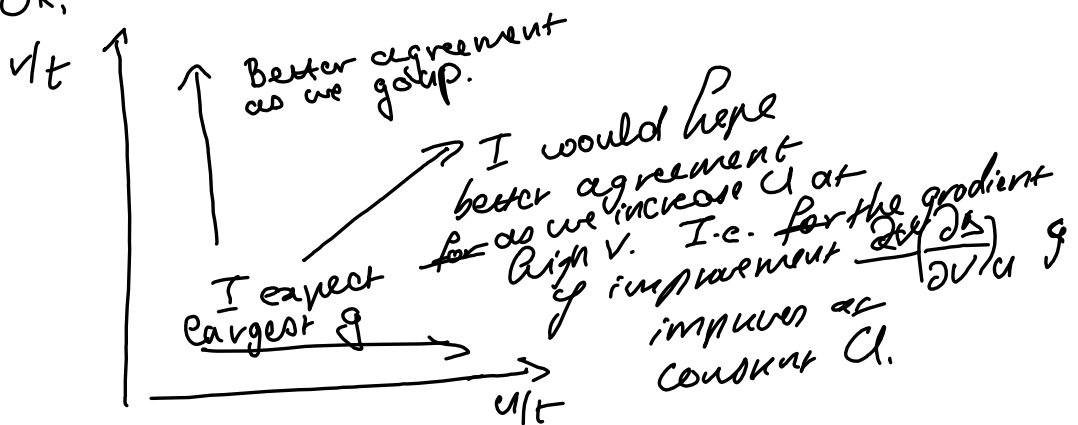
Hopefully you see it as within one clustering alone, but you may need to evaluate the ratio:

① matched cluster  $\propto \frac{(E_C - E_{\text{un}})/E_{\text{G}}}{(E_{\text{uc}} - E_{\text{cd}})/E_{\text{G}}}$   
 unmatched.

The key prediction is that we transition as a function of  $v/t$  to the matched case working better.

It would be good to understand analytically how this works - the projection is better?

OK, so I think:



That's a good question - ask for what  $\alpha$  you get the largest relative gap. How does that change as a function of  $V$ ?

OK. Key predictions

- ① As  $V_0$  increases, agreement should get better, matched case
- ② If not within matched case, then between matched and unmatched.

So, to do:

$$\textcircled{1} \frac{E_M - E_G}{E_G} \leftarrow \text{Goal for today.}$$

$$\textcircled{2} \frac{E_{UM} - E_G}{E_G} \leftarrow \text{Goal for next time.}$$

OK. So let's now try to architect the code for getting there.  
Let's start by just listing the ingredients:

- ① I need to make the clusters.  
↳ bonus: <sup>Do this in a way which</sup> allows for arbitrary nfrags.
- ② On each cluster I need to define the cluster Hamiltonian.
- ③ Solve for the spectrum at each cluster point.
- ④ Use the spectrum to find the occupation at each cluster.
- ⑤ Sum to find the total occupation.
- ⑥ Iterate this for different values of  $\mu_0$  so you can choose the filling. I.e. return the filling at constant  $\mu$ .  $\mu_0$  at a given filling.



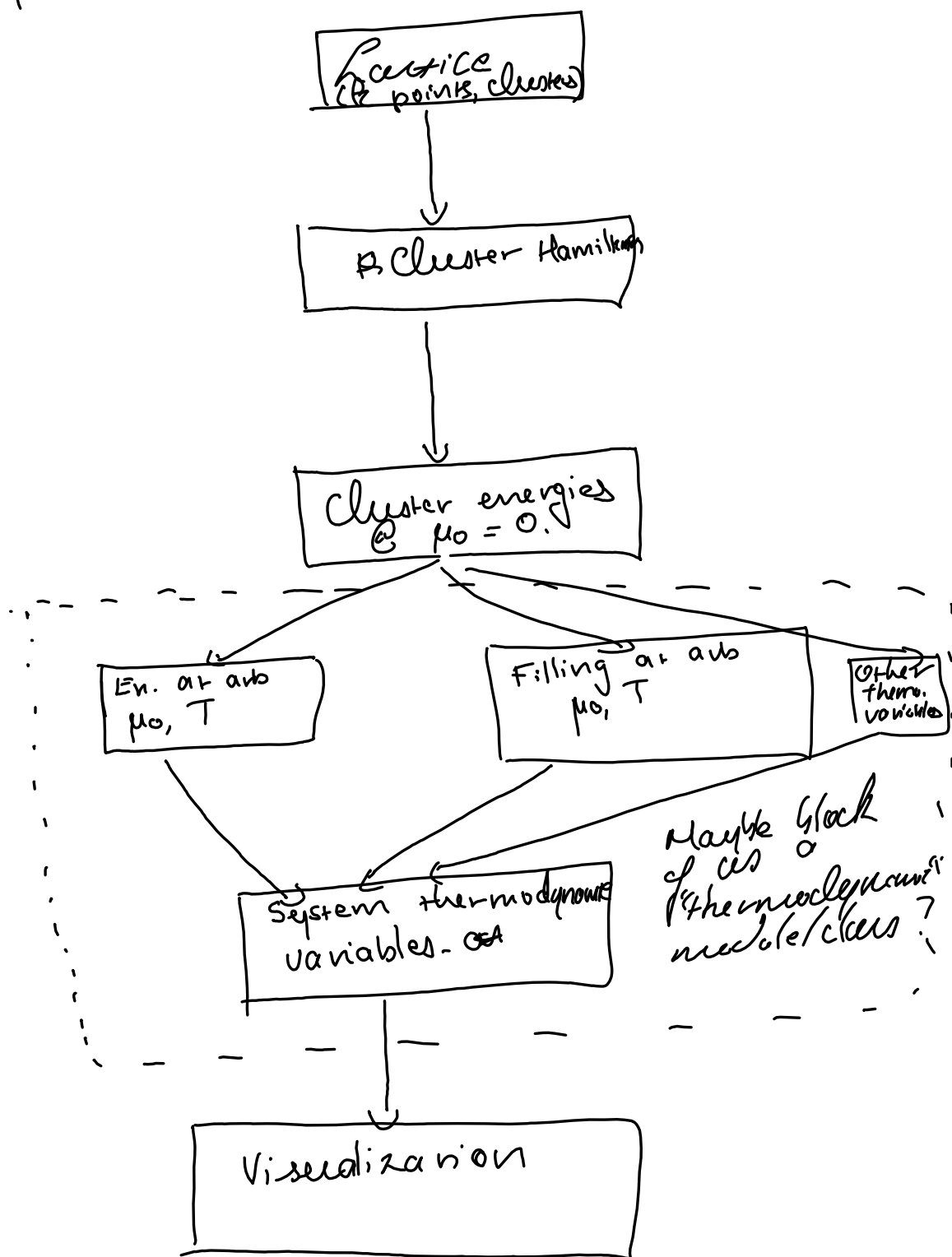
Remember that to find the filling for a new  $\mu_0$  you do not need to resolve the system. All you need to do is subtract the new amount and then evaluate the new average cluster occupation through:

$$\langle n \rangle = \frac{\sum_{\alpha} e^{-\beta(\epsilon_{\alpha} - \mu_0 N_{\alpha})} N_{\alpha}}{\sum_{\alpha} e^{-\beta(\epsilon_{\alpha} - \mu_0 N_{\alpha})}}$$

where  $\alpha$  indexes all possible states of the system.

OK. That seems right. The only thing that's a little unclear is how to deal with various  $\mu_0$ . I think the best way to do it is to solve first at  $\mu_0 = 0$  and then add  $\mu_0$  later.

Alright, so let's try to draw a flow diagram:



Note that it's at the Hamiltonian definition and solving level that I would input the DMRG. Since standard DMRG computes the energies and states system wide at zero-temp. I think it's probably best to keep the DMRG separate and then have a "handshake" module that ports over the thermodynamic variables.

the architecture hence gets modified  
to:

Lattice and  
system params.

Clusters

Define cluster  
 $H$

Get  $\mu_0 = 0$   
Full spectrum.

Fillings  
@  $\mu_0$

Energies  
@  $\mu_0$

Thermo.  
variables.

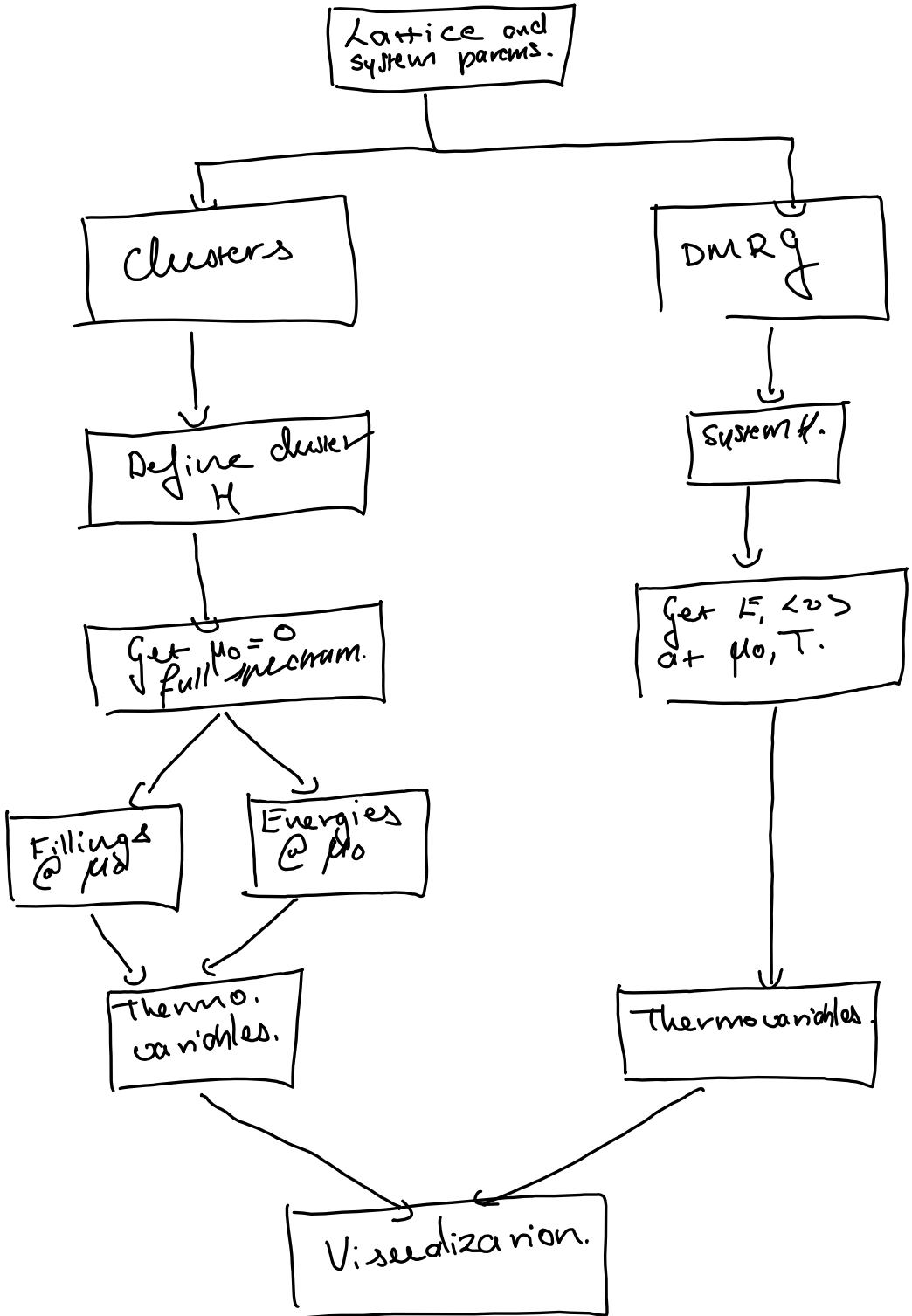
DMRG

System  $H$ .

Get  $E, \langle v \rangle$   
at  $\mu_0, T$ .

Thermovariables.

Visualization.





Can significantly improve DMRG runtime if you can easily evaluate thermodynamic variables at other  $\mu_0, T$  once you have the ~~exact~~ energies, states at some value, but I'm not sure this is possible.

This means you'll probably need a module to select a coarser  $\mu_0$  grid when searching for the fitting.