

The metal-insulator transition of a hydrogen chain using QWalk

Lucas K. Wagner

December 8, 2015

We will go through the simplest solid calculation that I can think of, a linear chain of hydrogen atoms, separated by a distance a . In order to make this calculation more tractable for an afternoon, we will make a few simplifying approximations:

- We will only consider 10 hydrogen atoms with periodic boundary conditions.
- Our single particle orbitals are constructed by hand using atomic orbitals

I've made a few Python scripts for you that will help construct the wave functions and so on. For most chemical systems, you'll use programs that convert from DFT codes to QWalk format. However, for simple enough systems, a script will do and is more convenient. We also typically use a script to generate trial wave functions for homogeneous electron gas calculations.

When setting up a QMC calculation, there are four main decisions to be made. The two most fundamental are the Hamiltonian and the observables of interest. These determine what physical questions will be addressed in the calculation. After those, the trial wave function and method should be chosen. We will walk through these choices for the hydrogen chain. Some of them will be motivated by efficiency—ideally you will be able to finish this tutorial within an afternoon.

For the hydrogen chain, we are going to be interested in the metal-insulator transition as the lattice constant a changes. One can probably understand that if the atoms are separated by a large distance a , then they will be insulating; they are too far for electrons to hop from one another. In the other limit, as a becomes small, it is not entirely obvious that the chain will become metallic. Actually most linear systems are susceptible to the so-called Peierls distortion, in which the system becomes dimerized and insulating for physically reasonable values of a . See polyacetylene for an example of a real system that does this. For different reasons, a 1D Hubbard model will always be insulating for $U \neq 0$. For the hydrogen chain, if we do not include dimerization, it will become metallic for a small enough.

Hamiltonian

We are going to do a simulation of a linear chain of hydrogen atoms. The Hamiltonian is:

$$\hat{H} = -\frac{1}{2} \sum_i \nabla_i^2 + \sum_{\vec{L}_i} \sum_i V(r_i) + \sum_{\vec{L}_i, \vec{L}_j} \sum_{ij} \frac{1}{\|\text{vecr}_i + \vec{L}_i - \vec{r}_j - \vec{L}_j\|}, \quad (1)$$

where i, j refer to electron indices, \vec{L} is summed over all periodic cells, and V is the one-particle potential we will define a little later. Units here and throughout this example are in *atomic units*, where $m_e = 1, \hbar = 1, e/4\pi\epsilon_0 = 1$. Energy in these units are called Hartrees, equal to 27.2114 eV, and length is in Bohrs, about 0.529 Å. In this form, the Hamiltonian applies to all systems of charged particles.

To define the chemical system, we must define $V(r_i)$. For our case, we will consider $N_H = 10$ hydrogen atoms in a periodic box, separated by distance a . $V(r_i)$ is thus given by

$$V(r_i) = - \sum_{\vec{L}} \sum_{\alpha}^{N_H} \frac{1}{|\vec{r}_i - \vec{r}_{\alpha} + \vec{L}|} \quad (2)$$

where α refers to the index of the hydrogen atom. This is the Coulomb interaction summed over an infinite lattice of our periodic images. This infinite sum is evaluated using the Ewald technique, which you will not have to worry about, since QWalk implements it automatically. All we need to tell the code is the \vec{r}_{α} values.

It is worth considering why we have to consider N_H atoms. Why not one or two? This is an interacting system, and so Bloch's theorem does not actually apply. One doesn't see this effect in DFT or Hartree-Fock calculations because they are *mean field* theories for which Bloch's theorem does apply. For the fully interacting simulation, we need to perform finite size scaling analysis, just as if we were performing a classical molecular dynamics simulation. For the purposes of the tutorial, we will consider a fixed size of N_H atoms, which I have determined has the same qualitative behavior as larger sizes. A complete analysis would check scaling to approach the true infinite system limit.

Observables

While there are many observables one could evaluate, we will focus on two main ones here. The first is the expectation value of energy:

$$E(\Psi_T) = \langle \Psi_T | \hat{H} | \Psi_T \rangle. \quad (3)$$

All the methods here are variational given a Hamiltonian, which means that $E(\Psi_T) \geq E(\Phi_0)$, where Φ_0 is the ground state. We thus often say that the best approximation to the ground state is given by the trial wave function with the lowest energy. This can be a useful guiding principle, and we will explore it for the metal-insulator transition.

The second observable is a two-particle probability distribution:

$$\rho_{\sigma_1, \sigma_2, r_1, r_2}(n_1, n_2), \quad (4)$$

where σ_i is the spin of each electron, and r_i refers to a 'region,' which by default is an atom. n_i is the number of electrons with spin σ_i nearest to atom r_i . This is a distribution, and so it contains a lot of information. Let's consider a few quantities we can calculate from this distribution.

First, let's look at the single-electron, single site properties,

$$\rho_{\sigma, \sigma, r, r}(n, n), \quad (5)$$

or just $\rho_{\sigma, r}(n)$. The average number of electrons of type σ on a site is

$$\langle n_{\sigma, r} \rangle = \sum_n n \rho_{\sigma, r}(n). \quad (6)$$

Its interpretation is fairly simple; this is the total charge nearest to a site.

Similarly, one can define the variance:

$$\text{var}(\sigma, r) = \langle (n_{\sigma, r} - \langle n_{\sigma, r} \rangle)^2 \rangle = \sum_n (n - \langle n_{\sigma, r} \rangle)^2 \rho_{\sigma, r}(n) \quad (7)$$

This is sometimes called the *charge compressibility*. For an atomic system, there is no hopping of electrons from one site to another, so this is zero, while for a metallic system, we expect this quantity to be larger, since electrons are hopping from site to site and so the number of electrons on a given site will fluctuate.

Finally, we can examine correlation properties of the wave function. We will mainly concern ourselves with the double occupation of the site:

$$\rho_{\uparrow, \downarrow, r, r}(1, 1). \quad (8)$$

This is the probability that site r has both an up and down electron present at the same time. For an atomic system, we expect this probability to be zero, since each atom either has one up or one down on it at a time. For a metallic, noninteracting system, we expect this to be approximately 0.25¹. For a correlated metallic system, this number will be something below 0.25, but decreases slowly with a , while the atomic system has an exponential decrease with increasing a .

Trial wave functions

Slater determinant

Throughout this work, we have discussed things in terms of the atomic limit and the non-interacting metallic limits. We will construct trial functions that reflect these limits, and see how their properties interact with the QMC methods. Let's call the $1s$ atomic orbital centered around atom i as

¹Question to the advanced reader: why is this not exactly 0.25? It would be for a non-interacting tight-binding model.

follows: $\chi_i(r)$. We will work only with this minimal basis. All of our wave functions will use a Slater determinant to enforce antisymmetry:

$$\Psi_S(r_1, r_2, \dots) = \text{Det}[\phi_j^\uparrow(r_k^\uparrow)] \text{Det}[\phi_j^\downarrow(r_k^\downarrow)], \quad (9)$$

where ϕ_j are the molecular/crystalline orbitals. In general,

$$\phi_j(r) = \sum_{\ell} c_{j\ell} \chi_{\ell}(r). \quad (10)$$

From the perspective of QMC, the main purpose of a DFT code is to calculate a good approximation to the coefficients $c_{j\ell}$. Again, we will construct them by hand for this simple system.

The atomic wave function (afm for antiferromagnetic) can be constructed out of $\chi_i(r)$ by

$$\phi_j^\uparrow(r) = \chi_{2j-1}(r) \quad (11)$$

$$\phi_j^\downarrow(r) = \chi_{2j}(r), \quad (12)$$

where j runs from 1 to $N_H/2$. This places an up electron on each odd atom and a down atom on each even atom. The non-interacting limit (tb for tight-binding) is given by solving the tight-binding Hamiltonian, and setting the $c_{i\ell}$'s accordingly.

Jastrow factor

We will introduce explicit electron correlations using a simple Jastrow factor:

$$\exp \left(\sum_{i,j,\alpha} u(r_\alpha, r_i, r_j) \right), \quad (13)$$

where remember that i, j are electron indices and α is a nuclear index. We will consider a small Jastrow, which is called *two-body* because it only includes terms between two particles at a time:

$$u(r_\alpha, r_i, r_j) = \sum_k c_k^{(en)} a_k(r_{i\alpha}) + \sum_k c_k^{(ee)} b_k(r_{ij}) \quad (14)$$

The electron-electron terms $c_k^{(ee)}$ and electron-nucleus terms $c_k^{(en)}$ will be optimized, and the basis functions a_k and b_k are covered in the QWalk documentation.

List of trial wave functions

We have four different trial wave functions: the two bare Slater determinants afm and tb, and two Slater-Jastrow wave functions afm-J and tb-J.

Methods

VMC: Monte Carlo evaluation of a trial wave function

We will use the VMC method to calculate the properties of each of our trial wave functions.

Variance optimization

We will use variance optimization (OPTIMIZE) to find the coefficients in the Jastrow.

DMC: diffusion Monte Carlo using a trial wave function

We will use the two Slater-Jastrow afm-J and tb-J as guiding functions and trial nodal surfaces for diffusion Monte Carlo.

Python scripts and running QWalk

- **gen_hchain.py** Set up the Hamiltonian, tb wave function, and afm wave function.
- **gen_method.py** Set up the Jastrow factor, the method parameters, and runqwalk(), which determines how to run QWalk on your system.
- **scan.py** Use functions in gen_hchain.py and gen_method.py to scan over parameter ranges.
- **read_numberfluct.py** Helper functions from qwalk/utils to read in the number probability densities.
- **gather.py** Convert the QWalk output files into a YAML database.

You will first want to modify gen_method.py, in particular the line (or similar) near the top:

```
def runqwalk(filename):  
    os.system("qwalk %s > %s.stdout" % (filename, filename))
```

which should correctly run the compiled qwalk executable on your system. If you compiled qwalk with MPI, you can insert the appropriate line, such as:

```
def runqwalk(filename):  
    os.system("mpirun -np 4 qwalk %s > %s.stdout" % (filename, filename))
```

which will speed up the execution.

You can use scan.py as a way to automate the calculations we need to do. Let's set up the calculation of the VMC(tb) and VMC(afm) calculations. You can do this as follows:

```
n=10  
avals= [1.0,2.0,3.0,4.0,5.0]  
for a in avals:  
    basename="h"+str(n)+str(a)  
    run_hf(basename,n,a)
```

Go ahead and run this program; it shouldn't take too long. To run the VMC(tb-J) and VMC(afm-J) calculations, do

```

n=10
avals= [1.0,2.0,3.0,4.0,5.0]
for a in avals:
    basename="h"+str(n)+str(a)
    run_variance(basename,n,a)

```

And finally, to get the DMC(tb=J) and DMC(afm-J) numbers,

```

n=10
avals= [1.0,2.0,3.0,4.0,5.0]
for a in avals:
    basename="h"+str(n)+str(a)
    run_dmc(basename,n,a)

```

While these calculations are running, you can go ahead with the next part; you'll just need to run `gather.py` every so often to update the database.

Saving the data

Finally, after performing all the calculations on all the trial wave functions, we will have the following levels of approximation:

- VMC(tb)
- VMC(afm)
- VMC(tb-J)
- VMC(afm-J)
- DMC(tb-J)
- DMC(afm-J)

These results are a function of the Hamiltonian as well, which as one parameter: the lattice constant a . Our results are thus functions of 3 parameters: trial wave function, method, a .

We will process this data by using a simple YAML database. `gather.py` will process the QWalk output file into a list of entries, each of which is a dictionary with the following keys:

a	Distance between hydrogen atoms (in Bohr)
twf	Trial wave function: one of tb or afm
method	Method used to evaluate results: one of vmc or dmc
energy	list of two numbers: energy[0] is the expectation value of the energy, energy[1] is the stochastic error bar
variance	Variance of the number of electrons on a site.
double_occupancy	The double occupancy on a site.

All observables have a corresponding stochastic error, `"_err."` It's very important to always plot this, and respect the stochastic errors! Many people have wasted a lot of time based on data that's too uncertain.

Plotting the data

There is an ipython notebook that will give you a starting point to plotting your results. It can group by the trial wave function or by the expected accuracy of the method (that is, VMC(slater), VMC(slater-jastrow), and then DMC(slater-jastrow)).

Some questions to answer:

- All these graphs should have a crossover between the tb and afm trial wave function obtaining lower energy. What happens when we add better treatment of the correlation? Why does that happen?
- How does the charge compressibility behave as a function of a ?
- What is the effect of correlation on the compressibility?
- How about the above two questions for the double occupancy?
- One measure of the quality of an initial guess is how much energy is gained upon adding the Jastrow and performing DMC. How does this behave as a function of a ?
- Another way to understand what was most incorrect about the trial wave function is to measure how much DMC changes quantities. Are there any regimes in which DMC changes one of our tracked quantities
- Where are our trial wave functions good? bad? Difficult question: why?
- More difficult question: how might we improve our wave function?