

Continuous-Time FCIQMC

August 19, 2010

The psip population in FCIQMC is evolved in imaginary time using a series of time steps, where each time step is of length τ . τ is set to be small to ensure convergence onto the lowest eigenstate of the Hamiltonian and so smaller values have to be used as the system size increases due to (typically) a larger spread in eigenvalues. A consequence of this is that the vast majority of attempted spawning steps are rejected, thus wasting a lot of CPU time. Here we describe how to perform FCIQMC without a time step by ‘jumping’ to the next spawning event.

We start from the standard diffusion equation used in FCIQMC, namely:

$$\frac{dc_{\mathbf{i}}}{dt} = - \sum_{\mathbf{j}} T_{\mathbf{ij}} c_{\mathbf{j}} \quad (1)$$

where

$$T_{\mathbf{ij}} = H_{\mathbf{ij}} - (\langle D_0 | \hat{H} | D_0 \rangle + S) \delta_{\mathbf{ij}}. \quad (2)$$

We define the matrices \mathbf{R} and $\mathbf{\Xi}$ by

$$R_{\mathbf{ij}} = |T_{\mathbf{ij}}| \quad (3)$$

$$\Xi_{\mathbf{ij}} = \text{sgn}(T_{\mathbf{ij}}) \quad (4)$$

and so the diffusion equation can be written as

$$\frac{dc_{\mathbf{i}}}{dt} = - \sum_{\mathbf{j}} \Xi_{\mathbf{ij}} R_{\mathbf{ij}} c_{\mathbf{j}}. \quad (5)$$

Consider a psip that is on determinant \mathbf{j} . The probability that this psip spawns a child onto determinant \mathbf{i} in time Δt is $R_{\mathbf{ij}} \Delta t$. Hence $R_{\mathbf{ij}}$ is the *rate* at which a psip on determinant \mathbf{j} spawns children on determinant \mathbf{i} . A psip on determinant \mathbf{j} may spawn children on all other determinants $\{\mathbf{i}\}$ (where $\{\mathbf{i}\}$ includes \mathbf{j} , i.e. spawning onto the same determinant is permitted), with each spawning connection occurring at its own rate.

What is the probability, $p_{\mathbf{i}}$, that the first child spawned appears on determinant \mathbf{i} at time $t = n\Delta t$?

$$\begin{aligned}
p_{\mathbf{i}} = & \underbrace{\prod_{\mathbf{k}} (1 - R_{\mathbf{kj}} \Delta t)}_{\text{prob. no children spawned in 1st time step}} \underbrace{\prod_{\mathbf{k}} (1 - R_{\mathbf{kj}} \Delta t)}_{\text{prob. no children spawned in 2nd time step}} \dots \\
& \times \underbrace{\prod_{\mathbf{k}} (1 - R_{\mathbf{kj}} \Delta t)}_{\text{prob. no children spawned in } (n-1)\text{-th time step}} \underbrace{\prod_{\mathbf{k} \neq \mathbf{i}} (1 - R_{\mathbf{kj}} \Delta t)}_{\text{prob. no children spawned on determinants } \mathbf{k} \neq \mathbf{i} \text{ in } n\text{-th time step}} \quad (6)
\end{aligned}$$

$$\begin{aligned}
& \times \underbrace{R_{\mathbf{ij}} \Delta t}_{\text{prob. child spawned on determinant } \mathbf{i} \text{ in } n\text{-th time step}} \\
& = \prod_{\mathbf{k}} (1 - R_{\mathbf{kj}} \Delta t)^n \left(\frac{R_{\mathbf{ij}} \Delta t}{1 - R_{\mathbf{ij}} \Delta t} \right) \quad (7)
\end{aligned}$$

$$\approx \prod_{\mathbf{k}} \left(1 - \frac{R_{\mathbf{kj}} t}{n} \right)^n R_{\mathbf{ij}} \Delta t \quad (8)$$

where the last line holds for sufficiently small Δt . Using the limit definition of the exponential function,

$$e^x = \lim_{n \rightarrow \infty} \left(1 + \frac{x}{n} \right)^n, \quad (9)$$

this becomes:

$$p_{\mathbf{i}} = \prod_{\mathbf{k}} \exp(-R_{\mathbf{kj}} t) R_{\mathbf{ij}} \Delta t \quad (10)$$

$$= \exp \left(- \sum_{\mathbf{k}} R_{\mathbf{kj}} t \right) R_{\mathbf{ij}} \Delta t. \quad (11)$$

Thus

$$\begin{aligned}
& p(\text{1st spawning is on determinant } \mathbf{i} \text{ between times } t \text{ and } t + dt) \quad (12) \\
& = e^{-R_{\mathbf{j}} t} R_{\mathbf{ij}} dt,
\end{aligned}$$

where we define $R_{\mathbf{j}} = \sum_{\mathbf{k}} R_{\mathbf{kj}}$. Summing over final states \mathbf{i} gives:

$$p(\text{1st spawning is between times } t \text{ and } t + dt) = e^{-R_{\mathbf{j}} t} R_{\mathbf{j}} dt. \quad (13)$$

We note that the probability is normalised:

$$\int_0^\infty R_{\mathbf{j}} e^{-R_{\mathbf{j}} t} dt = 1. \quad (14)$$

We can thus evolve the population dynamics by ‘jumping’ to the next spawning event. For each psip (on, say, determinant \mathbf{j}):

1. Pick next spawning time from the probability distribution function $f_t(t) = R_{\mathbf{j}}e^{-R_{\mathbf{j}}t}$. A point can be sampled from $f_t(t)$ as follows:
 - a) Choose a random number, u , from the uniform distribution on $[0, 1)$. $f_u(u)du = du$ gives the probability that u lies between u and $u + du$.
 - b) Calculate t from

$$t = -\frac{1}{R_{\mathbf{j}}} \ln(u) \quad (15)$$

so that $0 \leq t < \infty$.

Then as $f_t(t)|dt| = f_u(u)|du|$, it follows that $f_t(t) = R_{\mathbf{j}}e^{-R_{\mathbf{j}}t}$ as required.

2. Once the spawning time, t , has been chosen, choose the determinant \mathbf{i} on which to spawn according to

$$p_{\mathbf{i}} = \frac{R_{\mathbf{ij}}}{R_{\mathbf{j}}}. \quad (16)$$

This can be done simply by evaluating all the non-zero $p_{\mathbf{i}}$ values and using them to tile the interval $[0, 1]$. If a random number chosen from the uniform distribution on $[0, 1]$ lies on segment \mathbf{i} , then the spawning event occurs on determinant \mathbf{i} .

This process makes it possible to perform FCIQMC without time steps and no rejections, hence leading to efficient sampling. In the Hubbard model the summations $R_{\mathbf{i}}$ are easy and fast to evaluate.

However, each spawning event happens at different times for each psip. How can annihilation (which is crucial to avoid an exponential growth in noise) occur? We can do this by introducing ‘annihilation barriers’.

- Choose an annihilation time step, t_a . Since annihilation is costly, it might be sensible to make this reasonably large. However, if it is too large then the memory demands are that much greater and (more importantly) the noise in the simulation increases dramatically. A time step such that around 40 – 50% of psips spawn between annihilation barriers might be a good value.
- Advance all walkers, spawning event by spawning event, until the next event for that psip happens *after* t_a . As soon as this happens, advance the psip to time t_a but do not carry out the spawning event which happens after t_a . Note: we also have to advance all children to the

annihilation barrier (and the grandchildren and great-grandchildren etc.), which leads to an increase in memory as t_a increases. If it is easy to annihilate pairs of psips during the time evolution (e.g. if a psip spawns a child of the opposite sign on its own determinant) then do so.

- Once all psips in the simulation have reached t_a , annihilation is performed in exactly the same fashion as in standard FCIQMC.
- After annihilation, start advancing all remaining walkers to the next annihilation barrier at time $2t_a$.

Why does introducing annihilation barriers work? First, consider a psip on determinant \mathbf{j} at time $t < t_a$. If we ignore the annihilation barrier for a moment, then the probability of the next spawning event taking place between times t' and $t' + dt'$, where t' is after the annihilation barrier (i.e. $t' > t_a$) is simply

$$R_{\mathbf{j}} e^{-R_{\mathbf{j}}(t'-t)} dt'. \quad (17)$$

With the barrier present, the probability that the next spawning event occurs during the interval $[t', t' + dt']$ is given by:

$$\begin{aligned} \left(\begin{array}{l} \text{probability we} \\ \text{pick a time} \\ t_1 > t_a, \text{ so ad-} \\ \text{vancing to the} \\ \text{barrier} \end{array} \right) & \times \left(\begin{array}{l} \text{probability the} \\ \text{first event after} \\ \text{the barrier hap-} \\ \text{pens during the} \\ \text{interval}[t', t' + dt'] \end{array} \right) \\ & = \left(\int_{t_1=t_a}^{\infty} R_{\mathbf{j}} e^{-R_{\mathbf{j}}(t_1-t)} dt_1 \right) \times \left(R_{\mathbf{j}} e^{-R_{\mathbf{j}}(t'-t_a)} dt' \right) \end{aligned} \quad (18)$$

$$= e^{-R_{\mathbf{j}}(t_a-t)} R_{\mathbf{j}} e^{-R_{\mathbf{j}}(t'-t_a)} dt' \quad (19)$$

$$= e^{-R_{\mathbf{j}}(t'-t)} dt'. \quad (20)$$

Thus the introduction of the annihilation barrier has no effect on the statistics of the sequence of spawning events!

A Evaluation of $R_{\mathbf{ij}}$

An efficient continuous-time FCIQMC algorithm requires efficient enumeration of all the possible excitations from determinant \mathbf{j} and the associated $R_{\mathbf{ij}}$ matrix elements. Fortunately, this is relatively straight-forward in the Hubbard model (and also in, e.g., the uniform electron gas).

A.1 Real space spin-orbitals

If the determinant space is constructed from the real space (atomic) spin-orbitals, then only single excitations are connected and $R_{\mathbf{ij}}$ has only three possible values:

$$R_{\mathbf{ij}} = \begin{cases} R_{\mathbf{jj}} & \mathbf{i} = \mathbf{j} \\ t & |D_{\mathbf{i}}\rangle = a_a^\dagger a_i |D_{\mathbf{j}}\rangle, \text{ where } i \text{ and } a \text{ are respectively occupied and unoccupied spin-orbitals on neighbouring sites} \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

Due to the form of the Hubbard Hamiltonian, the maximum number of excitations possible from a determinant is $2n_{\text{dim}}N$, where n_{dim} is the number of dimensions in the system and N is the number of electrons. The number of possible excitations is often reduced by orbital a already being occupied.

A.2 Bloch spin-orbitals

In contrast, if the determinant is constructed from the Bloch spin-orbitals, then only double excitations are connected. Again, $R_{\mathbf{ij}}$ has three possible values:

$$R_{\mathbf{ij}} = \begin{cases} R_{\mathbf{jj}} & \mathbf{i} = \mathbf{j} \\ U & |D_{\mathbf{i}}\rangle = a_a^\dagger a_b^\dagger a_i a_j |D_{\mathbf{j}}\rangle, \text{ where } i, j \text{ and } a, b \text{ are respectively occupied and unoccupied spin-orbitals and conserve crystal momentum.} \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

Crystal momentum is conserved if

$$\mathbf{k}_i + \mathbf{k}_j = \mathbf{k}_a + \mathbf{k}_b, \quad (23)$$

where here \mathbf{k}_i refers to the wavevector of the i -th spin-orbital.

Due to the form of the Hubbard Hamiltonian, only double excitations where i and j are of opposite spin are allowed and the selection of the fourth spin-orbital involved in a connected excitation is uniquely determined by the other three spin-orbitals. Thus the maximum number of excitations from a determinant is $N_\uparrow N_\downarrow \min(M - N_\uparrow, M - N_\downarrow)$, where N_α is the number of electrons of spin α and $2M$ is the total number of spin-orbitals.