Quantum Monte Carlo Methods.

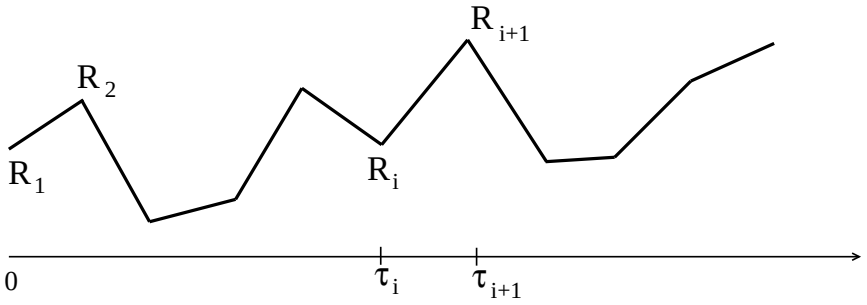
There are a number of ways how quantum simulations are done. Some directly solve the Schrodinger Eq. in imaginary time making use of the analogy with the diffusion equation and random walks. Some work with the Feynman diagrams, some use the path-integral (PI) representaion. I will start from the last one.

Path-integral Monte Carlo (PIMC) for continuous models.

So, we start from the expression derived in the last Sec.

$$Z = \oint \mathcal{D}\mathbf{R} \exp\left\{-\int_0^\beta \left[\sum_{i=1}^N m_i \dot{r}_i^2(\tau)/2 + \sum_{i=1}^N U(r_i(\tau)) + \frac{1}{2}\sum_{i\neq j}^N V(r_i(\tau) - r_j(\tau))\right] d\tau\right\} . \tag{1}$$

We immediately see that it is not suitable yet for the computer based solution because continuous trajectories require, by definition, and infinite set of numbers to describe them. To deal with this, we simply go one step back in the derivation, refuse to take the limit of $d\tau \to 0$, and keep it finite. Now each particle trajectory is described by a finite collection of points in imaginary time R_i



and system propagation between the points is approximated by staright lines.

The good news is that this description remains exact (!!) for the kinetic part of the action. Indeed, we have a formal identity

$$A_{R_1,R_2}(t) = \int dR_3 \; A_{R_1,R_3}(t_3)\, A_{R_3,R_2}(t - t_3) \; , \quad \text{for any} \;\; 0 < t_3 < t \tag{2}$$

which reads that propagation from R_1 to R_2 in time t over all possible paths is identical to the propagation from R_1 to R_2 in time t_3 over all possible paths

times the propagation from R_3 to R_2 in the remaining time $t - t_3$ with the integral taken over all possible points R_3. This immediately follows from the simple exponential form of A, and the fact that in both cases we talk about the same set of paths at the end. Thus

$$Z = \int \dots \int (dR_0\, dR_1\, \dots dR_L)\, \delta(R_0 - R_L) \prod_{i=0}^{L-1} A_{R_i,R_{i+1}}(\Delta\tau)\ ,$$

is an exact relation, no matter that L and $\Delta\tau$ are finite and arbitrary. Of course, we are not moving anywhere yet because A factors remain to be calculated. However, we may take advantage of the small value of $\Delta\tau$ and calculate $A_{R_i,R_{i+1}}(\Delta\tau)$ approximately, say to accuracy $\sim (\Delta\tau)^2$. It turns out that for the **free particle** the full path-integral expression

$$A_{0,R}(\tau) = \int \mathcal{D}\mathbf{R}\, \delta(R(0) - 0)\, \delta(R(\tau) - R)\ \exp\Big\{ -\int_0^\tau m\dot{R}^2(s)/2\, ds \Big\}\ ,$$

gives identically the same answer as the substitution of a linear path $R_{linear}(s) = (s/\tau)\, R$ into the exponent

$$A_{0,R}(\tau) \sim \exp\Big\{ -\int_0^\tau m\dot{R}^2_{linear}(\tau)/2 \Big\} = e^{-mR^2/2\tau}\ . \tag{3}$$

(I remind you that I ignore all prefactors which are supposed to cancel in the final averages). This fact can be established by direct integration, or simply by going back to the Schrodinger Equation and noticing that in imaginary time it is identical to the diffusion equation and thus (3) is the famous random walk distribution.

We see, in particular, that no approximations are done when we write for the free particle

$$Z^{(0)} = \int \dots \int dR_0\, dR_1\, \dots dR_L\, \delta(R_0 - R_L) \exp\Big\{ -\sum_{i=0}^{L-1} m(R_i - R_{i+1})^2/2\Delta\tau \Big\}\ .$$

This expression is already suitable for the MC simulation because we have a standard setup with the well-defined notion of the configuration and its positive definite weight. For interacting particles the potential energy exponent calculated for the linear trajectory

$$\int_{\tau_i}^{\tau_{i+1}} d\tau U[R_{linear}(\tau)]\ ,$$

is definitely an approximate answer, but for small $\Delta\tau$ the particle is unlikely to be found far from the linear path which is connecting nearby points in space; typical $|R_i - R_{i+1}| \sim \sqrt{\Delta\tau}$. For the single particle potential we may then write (a and b below enumerate particles)

$$S^{(a)}(i) = \int_{\tau_i}^{\tau_{i+1}} ds\, U_a[r^{(a)}(s)] = \int_{-\Delta\tau/2}^{\Delta\tau/2} ds\, U_a[\bar{r}_i^{(a)} + \bar{v}_i^{(a)} s] \; ,$$

and, after expanding U in Taylor series assuming small $\bar{v}_i^{(a)} s$, get

$$S^{(a)}(i) = U_a[\bar{r}_i^{(a)}]\Delta\tau + \frac{1}{3}(\bar{v}_i^{(a)})_\alpha (\bar{v}_i^{(a)})_\beta (U_a)''_{\alpha\beta}[\bar{r}_i^{(a)}]\Delta\tau^3 \; .$$

Here the following notations were used for the linear trajectory "center of mass" and "velocity" in order to simplify expressions, $\bar{r}_i^{(a)} = (r_{i+1}^{(a)} + r_i^{(a)})/2$ and $\bar{v}_i^{(a)} = (r_{i+1}^{(a)} - r_i^{(a)})/\Delta\tau$. Similarly, for each pairwise term of the interaction potential between particles we find

$$S^{(ab)}(i) = \int_{\tau_i}^{\tau_{i+1}} ds\, V[r^{(a)}(s) - r^{(b)}(s)] = \int_{-\Delta\tau/2}^{\Delta\tau/2} ds\, V[\bar{r}_i^{(a)} - \bar{r}_i^{(b)} + (\bar{v}_i^{(a)} - \bar{v}_i^{(b)})s] \; ,$$

and further on

$$S^{(ab)}(i) = V[\bar{r}_i^{(a)} - \bar{r}_i^{(b)}]\Delta\tau + + \frac{1}{3}(\bar{v}_i^{(a)} - \bar{v}_i^{(b)})_\alpha (\bar{v}_i^{(a)} - \bar{v}_i^{(b)})_\beta V''_{\alpha\beta}[\bar{r}_i^{(a)} - \bar{r}_i^{(b)}]\Delta\tau^3 \; .$$

Sorry for so many subscripts and superscripts but I would like to be specific about the time slice and particle index. Since velocities are of order $\Delta r/\Delta\tau \sim 1/\sqrt{\Delta\tau}$, the second term scales as $\Delta\tau^2$. We may keep expanding potential energy in the Taylor series to calculate the contribution of the linear path to higher order in $\Delta\tau$, but then we have to consider corrections for the non-linear path as well. This becomes somewhat messy and in most cases ignoring higher-order terms gives a reasonable accuracy. Thus we end up with the configuration weight exponent

$$-\sum_i^{L-1} \left\{ \sum_{a=1}^{N} \left(\frac{m_a (\bar{v}_i^{(a)})^2}{2} \Delta\tau + S^{(a)}[\bar{r}_i^{(a)}, \bar{v}_i^{(a)}] \right) + \sum_{a<b}^{N} S^{(ab)}[\bar{r}_i^{(a)}, \bar{r}_i^{(b)}; \bar{v}_i^{(a)}, \bar{v}_i^{(b)}] \right\} \; . \tag{4}$$

which is suitable for the simulation.

Small Appendix. In the **pair approximation** we first transform the path-ingtegral as (I assume, for simplicity, zero external potential, $U = 0$, and skip mentioning some of the indeces, e.g. in the integral limits)

$$A_{R_i,R_{i+1}}(\Delta\tau) = \int \mathcal{D}\mathbf{R} e^{-\int \Big(K[R(s)]+V[R(s)]\Big)ds} \equiv A^{(0)}_{R_i,R_{i+1}}(\Delta\tau)\ \langle e^{-\int dsU[R(s)]}\rangle_{FP}$$

where, by definition

$$A^{(0)}_{R_i,R_{i+1}}(\Delta\tau) = \int \mathcal{D}\mathbf{R}\, e^{-\int dsK[R(s)]} = e^{-\sum_{a=1}^{N} m_a(\bar{v}_i^{(a)})^2/2\Delta\tau}\ ,$$

is the free particle propagator, and $\langle \ldots \rangle_{FP}$ is defined as an average over trajectories with the free particle weight

$$\langle O[R(s)]\rangle_{FP} = \frac{\int \mathcal{D}\mathbf{R}\, e^{-\int dsK[R(s)]}\, O[R(s)]}{\int \mathcal{D}\mathbf{R}\, e^{-\int dsK[R(s)]}}\ .$$

Next, the average of the potential energy exponent is written approximately as the product of pairwise terms

$$\langle\ e^{-\int dsU[R(s)]}\ \rangle_{FP} \approx \prod_{a<b} \langle\ e^{-\int dsU[r^{(a)}(s)-r^{(b)}(s)]}\ \rangle_{FP}\ .$$

The advantage is that pairwise averages may be calculated either exactly (in many cases) or very precisely numerically. They may be also reduced to the solution of the Schrodinger equation for one particle (in the center of mass frame) in the central potential. I refer you here to (Storer, 1968) and (Pollock and Ceperley, 1984) for more details. These solutions are used to make more accurate expressions for the configuration weight (4).

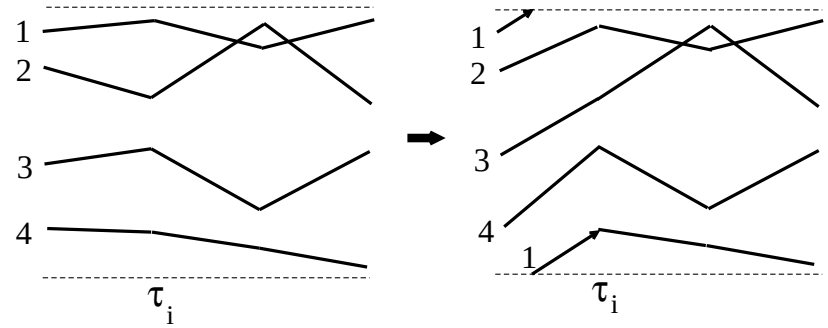
In the simplest PIMC scheme, just select time slice and particle at random and suggest a new value for the point $r_i^{(a)}$, e.g. with the Gaussian distribution function

$$\rho(r_i^{(a)}) \sim \exp\left\{-\frac{m}{\Delta\tau}\left[r_i^{(a)} - \frac{r_{i-1}^{(a)} + r_{i+1}^{(a)})}{2}\right]^2\right\}\ .$$

Then, accept the update using standard rules and the detailed balance Eq. For the procedure I am proposing, R is given by the ratio of the potential energy exponents in (4) only because kinetic energy terms cancel with the ratio of probability densities used for suggesting a new value of $r_i^{(a)}$ (this is the reason why it was done this way!).

In addition, when trajectories of two particles, say a and b, are within the specified distance, say $l = \#\sqrt{\Delta\tau/m}$, you may suggest two swap the values of particle coordinates, $r_i^{(a)} \to r_i^{(b)}$ and $r_i^{(b)} \to r_i^{(a)}$ to facilitate more efficient "entanglement" of trajectories. Now $R = e^{-\Delta S}$ where ΔS is the action difference calculated using Eq. (4).

Still, winding numbers may not be introduced this way. If system size is small, and no tricks are used, then winding number may be introduced by shifting cordinates of **many** particles at once in the left or in the right sence with equal probabilities. The illustration below is for the one-dimentional case, but the idea is clear and may be implemented in a similar way (with more programming effort) in any dimension.



For the randomly selected time slice τ_i, and shift $g = \pm 1$ order particle coordinates and perform $R_i^{(a)\,\prime} = R_i^{(a+g)}$ (pay attention to the periodic boundary conditions; for $g = 1$ and $a = N$ use $a + g = 1$, for $g = -1$ and $a = 1$ use $a + g = N$ as in figure above). This update will change the winding number by g. The downside is that the acceptance ratio for this **macro-update** is exponentially small in large systems. I believe, the solution is in performing Worm Algorithm updates (not implemented yet, as far as I know).

Diagrammatic Monte Carlo

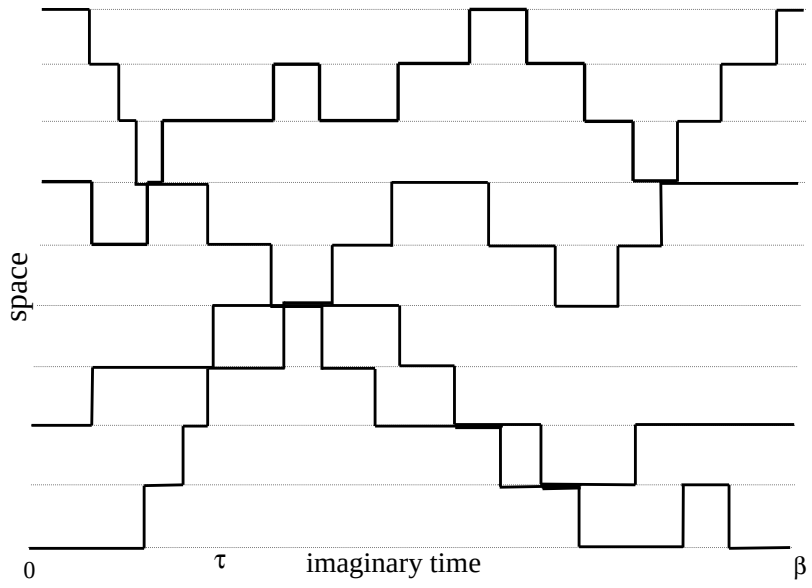
I will start from the particular example of quantum system which we already discussed, and then formulate it in general terms as a problem of summing

series of integrals. Next, I will show how to do Metropolis type MC simulation of such series without any approximations. After the general theory is developed, I will complete the loop and go back to consider quantum algorithms for the interacting many-body system on a lattice. The logic is similar to our general discussion of the MC approach to classical systems and its application to the Ising model.

To be specific, consider the bosonic Hubbard model introduced previously

$$H = -\Delta_0 \sum_{<ij>} (b_i^\dagger \, b_j + h.c.) + \frac{U}{2} \sum_i n_i^2 + \sum_i \xi_i n_i \ , \tag{5}$$

and, for convenience, reproduce also the typical structure of the configuration space



and the corresponding partition function

$$Z = \sum_{K=0}^{\infty} \int_0^\beta d\tau_1 \ldots \int_{\tau_{K-1}}^\beta d\tau_K \sum_{i_{CP}(\tau)} W_K[i_{CP}(\tau)] \tag{6}$$

with the configuration weight

$$W_K[i_{CP}(\tau)] = (\Delta_0)^K \ \exp\Big\{ -\sum_{a=1}^{K} \int_{\tau_{a-1}}^{\tau_a} \Big(\frac{U}{2} \sum_j n_{j(\tau)}^2 + \sum_j \xi_j n_{j(\tau)} \Big) \, .d\tau \Big\} \tag{7}$$

Here the sum over j goes over all lattice points, and $n_{j(\tau)}$ are occupation numbers on site j at time τ for the particular many-body trajectory $i(\tau)$ with K hopping transitions between the lattice sites.

The generic formulation of what we are doing is as follows:
Suppose one is interested in the quantity $A(y)$ defined as

$$A(y) = Z^{-1} \sum_{K=0}^{\infty} \int dx_1 \ldots \int dx_K \sum_{\xi_K} A(K, \xi_K; x_1, \ldots, x_K; y) W(K, \xi_K; x_1, \ldots, x_K; y) \tag{8}$$

$$Z = \sum_{K=0}^{\infty} \int dx_1 \ldots \int dx_K \sum_{\xi_K} W(K, \xi_K; x_1, \ldots, x_K; y) \tag{9}$$

where y stands for some external (continuous or discrete) parameters, K is the series index equal to the number of integrals over parameters x_i [in general, x_i are multidimensional, i.e. $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots)$], ξ_K enumerates different terms which have the same series index. The collection of numbers $(K, \xi_K; x_1, x_2, \ldots, x_N; y)$ specifies a particular configuration, or diagram, ν. Notice, that configurations=diagrams with different K have different number of continuous parameters to characterize them. In our example above, K is the number of transitions between the lattice sites ("kinks") in the many-body trajectory, x_i are imaginary times of these transitions, ξ_K is specifying exact locations in space for all kinks, their direction (in which direction the particle has moved between the sites), and initial occupation numbers on all sites—literally, we simply mention everything necessary to draw the trajectory evolution from some initial state. y parameters may refer, for example, to the Hamitonian parameters. I call configurations "diagrams" after Feynman, and because we have nice graphical diagrams (previous page) to visualize $(K, \xi_K; x_1, x_2, \ldots, x_N)$.

If we want to do the MC simulation of this series, we have to suggest a set of updates which allow to transform arbitrary configuration into each other. Updates which conserve the series index K, lets call them "**type-A**" updates, can be implemented using standard classical rules. For example, one can select at random any of the integration variables x_k and propose a new value for it. The acceptance ratio will be proportional to

$$R \sim \frac{W(K, \xi_K; x_1, \ldots, x'_k, \ldots, x_K; y)(dx)^K}{W(K, \xi_K; x_1, \ldots, x_k, \ldots, x_K; y)(dx)^K} ,$$

or

$$R \sim \frac{W(K, \xi_K; x_1, \ldots, x'_k, \ldots, x_K; y)}{W(K, \xi_K; x_1, \ldots, x_k, \ldots, x_K; y)} \ .$$

All differential measures cancel out because we do not change the integral multiplicity. Similarly, one may suggest to modify y parameter, or to go from ξ_K to ξ'_K. As long as we do not change K our job is no different from the standard classical simulation of the multidimensional integral.

"**Type-B**" updates suggest to change the value of K. We can not avoid them altogether because the answer has meaning only as a sum of many terms in the series, and *a priori* we do not know which terms in the series are dominating in the answer. In any case, ergodicity requires that all terms must be accounted for in the long run. If we pretend that dx are finite, then we may simply suggest to increase K by n by adding new variables $\{x\}_{new} = x_{K+1}, \ldots x_{K+n}$ (by going backwards we suggest to decrease K by n by removing $\{x\}_{new}$ variables). The acceptance ratio for B-updates

$$R \sim \frac{W(K+n, \xi_{K+n}; x_1, \ldots, x_K, x_{K+1}, \ldots x_{K+n}; y)(dx)^{K+n}}{W(K, \xi_K; x_1, \ldots, x_K; y)(dx)^K} \sim (dx)^n \ ,$$

contains now uncompensated differential measures. Of course, if dx is very small the acceptance ratio is small and the algorithm is inefficient. But what else can we do? Taking the limit $dx \to 0$ explicitly would result in $R = 0$ and impossibility to perform B-updates. For many years people believed that making dx finite is the only way to proceed.

The solution to the problem becomes clear if we ask the following question: What is the relative weight of the diagram $\nu_K = (K, \xi_K; x_1, \ldots, x_K; y)$ versus **all** diagrams of higher order $\nu'_{K+n} = (K+n, \xi_{K+n}; x_1, \ldots, x_K, x_{K+1}, \ldots x_{K+n}; y)$ with $\{x\}_{new} \in$(a certain n-dimensional volume)$=V^{(n)}_{new}$. The **integrated** weight of new diagrams (I will not mention all arguments of W below for brevity)

$$\int \ldots \int_{V^{(n)}_{new}} W(x_1, \ldots x_{K+n})(dx)^{K+n} = W^{(\mathbf{int})}(x_1, \ldots, x_K)(dx)^K \ , \qquad (10)$$

has the same number of differential measures as the original diagram. The acceptance ratio for the update suggesting to transform ν_K into *any* of the ν'_{K+n} with $\{x\}_{new} \in V^{(n)}_{new}$ is then finite

$$R \sim \frac{W^{(int)}(K+n, \xi_{K+n}; x_1, \ldots, x_K; y)}{W(K, \xi_K; x_1, \ldots, x_k, \ldots, x_K; y)} \ . \qquad (11)$$

Well, if this update is accepted, and we **have to** increase K by n then we may select the values of new variables $\{x\}_{new}$ within the volume $V_{new}^{(n)}$ using normalized(!!) probability density

$$\frac{W(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)}{W^{(int)}(K+n,\xi_{K+n};x_1,\ldots,x_K;y)}\ . \tag{12}$$

The net result is a finite acceptance ratio for transforming ν_K into ν_{K+n}.

So, what is the "miracle" here? It is a simple twist of imagination that we perform B-updates not as (K-diagram) *vs* (K+n-diagram) comparison, but as (K-diagram) *vs* locally integrated (K+n-diagrams). You may say, *"A-ha!, one needs to perform horrible integrals (10) to make this scheme work, and in addition to seed new variables $\{x\}_{new}$ with the horrible distribution function (12). This is where all the trouble is now!"* Since I am assuming that W functions are arbitrary, it may be a problem indeed to perform integrals numerically in each update. However, there is a simple trick around this problem based on re-weighing ideas, or the hibrid method discussed in the RNDM Section.

I will go directly to the detailed balance Eq. for the B-update to derive the final expression for the acceptance ratio. As before, p_u is the probability to apply a particular B-update transforming ν_K to ν'_{K+n} by adding n new variables, and $p_{\bar{u}}$ is the probability of going backwards and simply removing n variables from the diagram. To perform the B-update, a simple distribution function $P(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)$ is introduced to seed variables $x_{K+1},\ldots x_{K+n}$. This function should allow the possibility of using the transformation method in a closed analytic form, so that no CPU time is wasted on the normalization integral

$$\int\ldots\int_{V_{new}^{(n)}} P(x_1,\ldots x_{K+n})(dx)^{K+n} = P^{(int)}(x_1,\ldots,x_K)(dx)^K\ , \tag{13}$$

and the seeding process (again, not arguments are mentioned; I will suppress arguments in the P-functions below as well for simplicity). The detailed balance Eq. then reads

$$W(K,\xi_K;x_1,\ldots,x_K;y)(dx)^K\ p_u\ \frac{P(x_{K+1},\ldots x_{K+n})(dx)^n}{P^{(int)}}\ P_u^{acc}(\nu_K\to\nu'_{K+n}) =$$

$$W(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)(dx)^{K+n}\ p_{\bar{u}}\ P_{\bar{u}}^{acc}(\nu'_{K+n}\to\nu_K)\ ,$$

where factor-by-factor we mention the configuration weight, the update probability, the probability of suggesting a particular set of new variables, and the probability of accepting the update. All differential measures cancel in the balance Eq. and we get the acceptance ratio as

$$R=\frac{p_{\bar{u}}}{p_u}\ \frac{W(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)\ P^{(int)}}{W(K,\xi_K;x_1,\ldots,x_K;y)\ P(x_{K+1},\ldots x_{K+n})}\ . \qquad (14)$$

Of course, when $P=W(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)$ we reproduce our previous result (11). Formally, P is arbitrary (the final result may not depend on the choice), but the algorithm efficiency strongly depends on it. On one hand we want it to be simple enough to use the transformation method, on the other hand, we want it to be as close as possible to the $W(K+n,\xi_{K+n};x_1,\ldots,x_K,x_{K+1},\ldots x_{K+n};y)$ function to maximize the acceptance ratio.

We have essentially completed the theory of Diagrammatic MC because we know now how perform an ergodic set of updates on the series of integrals. An amazing result is that in the final scheme we keep summing numbers A_ν and pay no attention whatsoever to the fact that different diagrams have different number of continuous variables and the ratio of their weights is typically 0 or ∞ ! In a way, we sum series of integrals without taking any of the integrals explicitly!

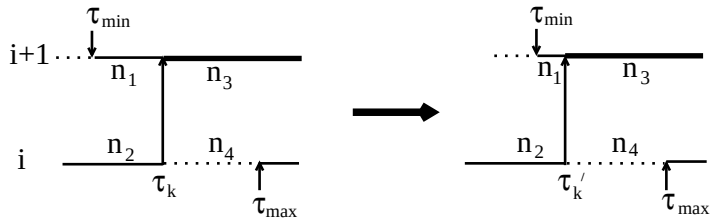
Using many-body lattice path-integral as an example of Diagrammatic MC.

This subsection is a simple illustration of how Diagrammatic MC can be applied to the lattice path-integral. I will describe the most straightforward scheme first, and postpone the discussion of the Worm-Algorithm for the same model.

So, let's go back to the bosonic Hubbard model, and consider its partition function as a particular case of the diagrammatic series. By looking at the trajectory below Eq. (5) one may suggest three updates: A-type **time**

shift, B-type **kink-antikink creation** and its counterpart **kink-antikink annihilation**.

Time shift (Type A). In this update one suggests to select at random any kink in the configuration and to change its imaginary time coordinate. In the picture below the selected kink is at time τ_k and connects sites i and $i+1$. It changes site occupation numbers from n_1 to $n_3 = n_1 + 1$ on site $i+1$, and from n_2 to $n_4 = n_2 - 1$ on site i. Without changing the nature of other kinks in the system it can be placed anywhere between times τ_{min} and τ_{max} (which are times of other kinks in the system).



We may propose the new position in time for the kink k using simple uniform probability density

$$\tau_k' = \tau_{min} + rndm()(\tau_{max} - \tau_{min}) \; .$$

The acceptance ratio for this choice will be given by the change of the potential energy only (the matrix element for the kink $\langle\, n_1, n_2|\, -K|\, n_3, n_4\rangle = \Delta_0\sqrt{n_3 n_2}$ remains the same)

$$R = \exp\left\{-\int_{\tau_k}^{\tau_k'} d\tau \left(\frac{1}{2}\sum_{jj'} U_{jj'}\Big[n_j'(\tau)n_{j'}'(\tau) - n_j(\tau)n_{j'}(\tau)\Big] - \sum_j \xi_j\Big[n_j'(\tau) - n_j(\tau)\Big]\right)\right\} ,$$

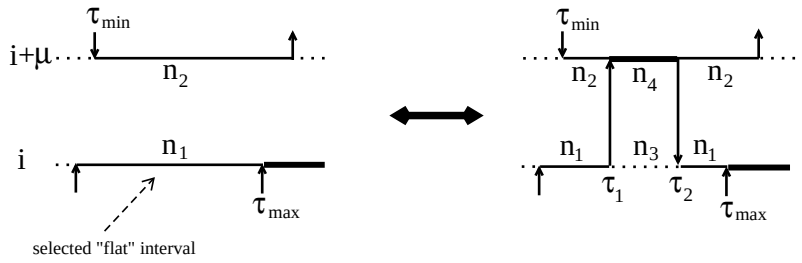
where occupation numbers in the final diagram are exactly same as in the initial one except sites i and $i+1$ where $n_{i+1}' = n_1$, $n_i' = n_2$, and $n_{i+1} = n_3$, $n_i = n_4$, It is written in the general form assuming arbitrary interaction potential between particles. For the Hubbard model with the on-site interaction potential it simplifies to

$$R = \exp\left\{-(\tau_k' - \tau_k)\left(\frac{1}{2}U(n_1^2 + n_2^2 - n_3^2 - n_4^2) - \xi_{i+1}(n_1 - n_3) - \xi_i(n_2 - n_4)\right)\right\} ,$$

At this point you would probably immediately say "Not uniform probability density! Use the heat bath idea and generate the new value of τ_k'

according to $u(\tau) = Ee^{-\tau E}/(e^{-\tau_{min}E} - e^{-\tau_{max}E})$ defined on $\tau \in (\tau_{min}, \tau_{max})$ where $E = U(n_1^2+n_2^2-n_3^2-n_4^2)/2-\xi_{i+1}(n_1-n_3)-\xi_i(n_2-n_4)$ is the potential energy change due to kink k. Your acceptance ratio will become unity!". I certainly agree with this approach.

Kink-antikink annihilation (Type B). In this update we select some "flat" part of the trajectory, i.e. the one which has no kinks on it, and suggest to modify it by creating a kink-antikink pair as shown in the figure below.



The selection process requires some care. We may enumerate, or count, all "flat" parts of the trajectory and select any of them at random. Let the number of "flat" be N_{fl}. In a well developed trajectory with many kinks between all sites we have $N_{fl} = 2K$, however, if there are sites which are not connected by kinks to any other site, then we have $N_{fl} < 2K$. One has to be careful in properly counting this number in type-B updates because it is part of the detailed balance Eq. below. So, we select the "flat" interval using $rndm() * N_{fl} + 1$. Next, we may select at random the direction in space to determine the n.n. site we are going to make a hopping transition, $\hat{\mu} = rndm() * d + 1$, and determine the imaginary time boundaries within which we will place a new kink-antikink pair. For simplicity of calculating the acceptance ratio, I suggest to determine τ_{min} and τ_{max} from the condition that the new kink-antikink pair does not interfere with any of the existing kinks. This is **not** obligatory though; other choices are possible too but they will require a laborous examination of how occupation numbers and the corresponding matrix elements for other kinks have changed. The stage for the update is ready now. I will no longer do, or mention, heat-bath improvements which can be easily done on top of simple updates.

We select at random $\tau_1 = \tau_{min} + rndm() * (\tau_{max} - \tau_{min})$ and $\tau_2 = \tau_{min} + rndm() * (\tau_{max} - \tau_{min})$, and order them so that $\tau_1 < \tau_2$. In other words, we select imaginary time positions for the kink-antikink pair with the uniform

probability density $u(\tau_1,\tau_2) = 2/(\tau_{max}-\tau_{min})^2$. The corresponding changes in the occupation numbers are $n_3 = n_1 - 1$ and $n_4 = n_2 + 1$; the product of matrix elements for the kink and for the antikink is $t^2 n_1(n_2+1)$. Note, that the number of flat intervals has increased, and is now $N'_{fl} > N_{fl}$.

When going backwards, we select at random a flat interval (out of N'_{fl}), verify that its imaginary time boundaries are due to the kink-antikink pair, i.e. due to the particle hopping to some site and and coming back from exactly same site (in the figure above the n_2, n_4 intervals do not qualify for the kink-antikink removal according to my rules), and if the removal of the kink-antikink pair is not interfering with other kinks, then we suggest to erase the pair. It is easy to write these rules, and make a plot explaning what they mean, but it takes some time to program everything carefully and make sure that every single exception is mentioned, to have the right data structure of kinks and connected sites, proper handling of the $\beta - periodic$ time variables, etc.

The detailed balance Eq. for the type-B update discussed above is

$$(\mathrm{Case})_\nu \, \frac{1}{N_{fl}} \, u(\tau_1,\tau_2)(d\tau)^2 \, W_\nu (d\tau)^K \, P^{(acc)}(\nu \to \nu')$$

$$= (\mathrm{Case})_{\nu'} \, \frac{1}{N'_{fl}} \, W_{\nu'} (d\tau)^{K+2} \, P^{(acc)}(\nu' \to \nu) \; ,$$

which gives

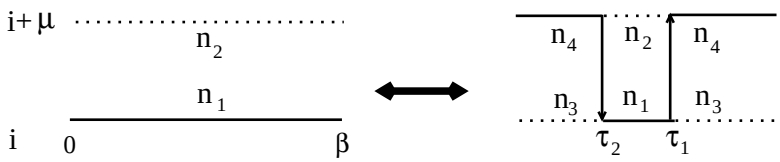
$$R = \frac{(\mathrm{Case})_{\nu'}}{(\mathrm{Case})_\nu} \, \frac{N_{fl}}{N'_{fl}} \, \frac{(\tau_{max}-\tau_{min})^2 W_{\nu'}}{2W_\nu} \quad .$$

For the Hubbard model the ratio of configuration weights is

$$\frac{W_{\nu'}}{W_\nu} = t^2 n_1(n_2+1) \, e^{-(\tau_2-\tau_1)\,[U(n_3^2+n_4^2-n_1^2-n_2^2)/2 - \xi_i(n_3-n_1) - \xi_{i+1}(n_4-n_2)]} \; .$$

The (Case) factor takes care of special cases when kink-antikink pairs are inserted on intervals which have no other kinks. In the picture below I show an example. It is clear, that there is no way of telling by looking at the final configuration whether the kink-antikink pair was created starting from the flat interval n_1 with $\tau_2 < \tau_1$ or interval n_2. (on the β-periodic circle $\tau_2 < \tau_1$ is allowed, which means that $u(\tau_1,\tau_2) = 1/\beta^2$). Thus, $(\mathrm{Case})_\nu = 2/2 = 1$. Correspondingly the pair can be deleted if either n_3 or n_2 interval is selected,

and thus $(\text{Case})_{\nu'} = 2$.



The last question I would like to discuss in connection with the lattice path-integral approach is an estimator for the kinetic energy. The estimator for the potential energy is easy and self evident

$$U_\nu = \beta^{-1} \int d\tau \frac{1}{2} \sum_{jj'} U_{jj'} n_j(\tau) n_{j'}(\tau)$$

The average kinetic energy $E_{kin} = -\Delta_0 \sum_{<ij>} (b_i^\dagger b_j + h.c.)$ can be obtained from

$$\langle E_{kin} \rangle = Z^{-1} \sum_a E_{kin} e^{-\beta(K+U)} \equiv \left(-\frac{\Delta_0}{\beta} \right) \; Z^{-1} \; \frac{\partial Z}{\partial \Delta_0} \; .$$

From Eqs. (6) and (7) we know that

$$Z = \sum_{K=0}^{\infty} \sum_{\nu_K} W_{\nu_K} \; ,$$

with $W_{\nu_K} \propto \Delta_0^K$ where ν_K are all possible trajectories with K kinks. Thus

$$\Delta_0 \frac{\partial Z}{\partial \Delta_0} = \sum_{K=0}^{\infty} \sum_{\nu_K} K \, W_{\nu_K} \; ,$$

and the estimator for the kinetic energy is simply

$$\langle E_{kin} \rangle = - < K > / \beta \; .$$

This concludes a simple MC scheme which samples lattice trajectories in a given winding number sector. Changing winding numbers can not be achieved in the local scheme. The WA scheme discussed next solves the problem of winding numbers. Also, WA has virtually no critical slowing down problem, can simulate grand canonical ensembles, disordered systems, etc,

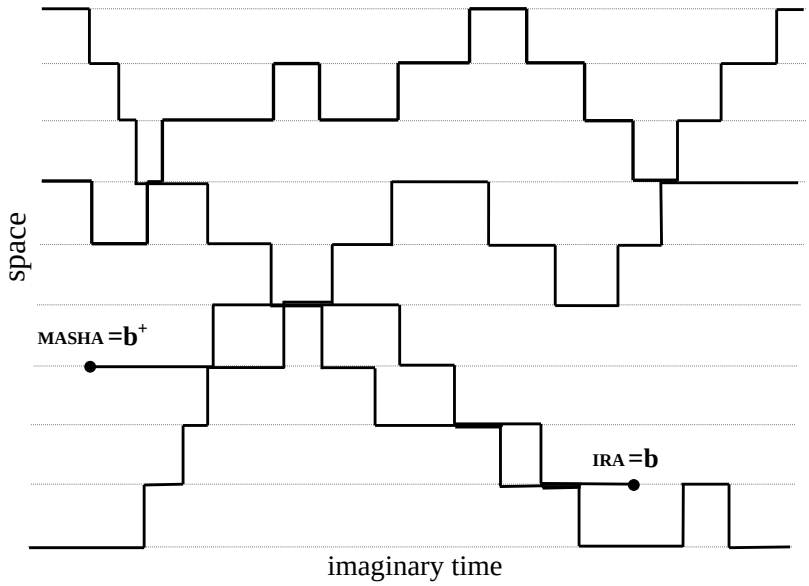
and, for no price at all, gives us the single particle propagator in imaginary time

$$G(j,\tau) = \langle \hat{b}_{i+j}(t+\tau)\ \hat{b}_i^\dagger(t) \rangle\ ,$$

where, by definition, the meaning of $\hat{b}_i^\dagger(t)$ is to create a particle on site i at time t with the matrix element $\sqrt{n_i(t)+1}$, and the meaning of $\hat{b}_{i+j}(t+\tau)$ is to delete a particle on site $i+j$ at time $t+\tau$ with the matrix element $\sqrt{n_{i+j}(t+\tau)}$. The only difference between the configurations contributing to the partition function Z and those contributing to G is that one of the trajectories starts at (i,t) and terminates at $(i+j,t+\tau)$.

Path-integral Monte Carlo for lattice models. Worm Algorithm.

So, our configuration space is now

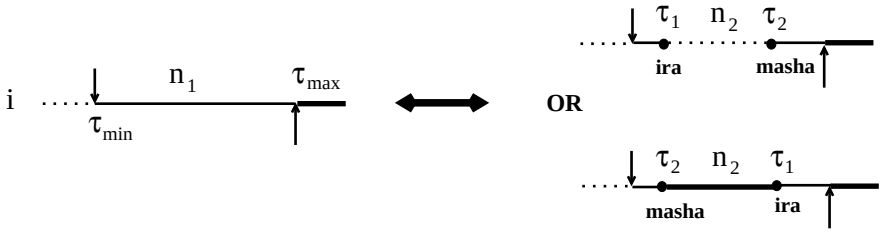


For historical reasons the $\hat{b}^\dagger$-point is called "Masha", and the $\hat{b}$-point is called "Ira" (Ira and Masha are the names of Boris Svistunov's sisters and integers in Fortran). The WA strategy is exactly same as before: draw trajectories using space-time moves of *ira* exclusively, and count each move as a contribution to the Green function $G(j_{ira}-j_{masha}, \tau_{ira}-\tau_{masha})$ histogram. I will be short in describing all the details of updates and their obvious improvements

(they are very much the same as discussed above), and underline only new features.

To switch between the Z and G configuration spaces we use a complimentary pair of updates which **create/delete** $ira - masha$ pair on the same flat interval.

• In **create** (this update is used only if we are in the Z-configuration space; you may introduce a logical variable $present = .false.$ to tell you if this is the case) we select at random one of the flat intervals, and in terms of the figure below we suggest to place ira at time $\tau_1 = \tau_{min} + rndm() * (t_{max} - \tau_{min})$, and $masha$ at time $\tau_2 = \tau_{min} + rndm() * (t_{max} - \tau_{min})$.



• In **delete** (this update is used only if we are in the G-configuration space, i.e. $present = .true.$ and ira and $masha$ are the ends of the same flat interval) we suggest to erase $ira - masha$ pair from the configuration. Just read the figure above backwards. Notice, that other updates are possible when ira and $masha$ are present, so the delete-update is applied with some probability p_d. The detailed balance Eq. for the create/delete updates reads

$$(\mathrm{Case})_\nu \, \frac{1}{\mathrm{N_{fl}}} \, \mathrm{W}_\nu \, (\mathrm{t_{max}} - \tau_{\mathrm{min}})^2 \, \mathrm{P_c^{(acc)}} = (\mathrm{Case})_{\nu'} \, \mathrm{p_d} \, \omega_{\mathrm{G}} \, \mathrm{W}_{\nu'} \, \mathrm{P_d^{(acc)}} \ ,$$

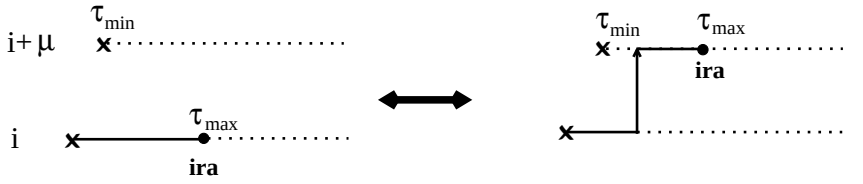
(as before, ω_G is a free parameter $\sim \langle 1/N_{fl} \rangle$).

To develop the configuration and update its structure we need updates shifting ira in time and in space.

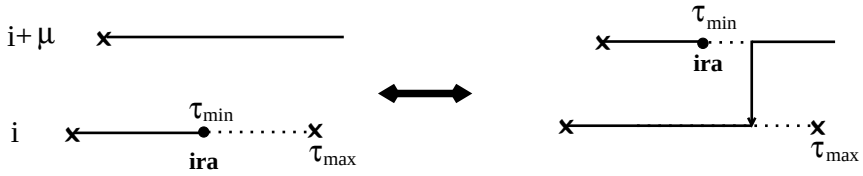
• In the **time shift** update we perform an action similar to the kink shift update discussed above. Just a figure is enough to explain it. This update

is used only if $present = .true.$ and its probability is p_{ts}.

• In the **space shift** updates we suggest to create/delete a kink to left or to the right (forward or backward in time) of ira. The probability of calling any of the space shift updates is p_{ss}. Again, I will simply show a figure of what space shift updates do (the τ_{min} and τ_{max} limits are decided on the basis of the current ira position in time and the requirement that the kink to be created/deleted does not interfere with any of the existing kinks)



Space Shift (left)



Space Shift (right)

This is a complete set of updates which do all the necessary job. You can draw any line, erase any line, and jump between the lines, see "Space Shift (right)" figure. Winding numbers in space and time are not a problem at all. Green's function is the "by-product" of the algorithm, but its space-time histogram allows you to get correlations in the system, spectrum of elemenraty exitations, etc.

Optimization problems and Variational MC.

This is a sudden change in the line of approach to the quantum system. In fact, I would like to discuss first how one can do optimization problem using random numbers. Suppose we have a function of many variables $f(c_1, c_2, \ldots, c_N)$ which is so complex that there is no simple way of finding its global minimum. Moreover, it may have many local minima, so that a simple speepest-descent method = "go in the direction of ∇f" does not work, because it will stuck at some local minimum. Optimization using random numbers may go as follows:
1. Initialize the calculation by setting $f_{min} = 1.d200$, i.e. as large as possible.
2. Choose at random a set of parameters $(c_1, c_2, \ldots, c_N)$, e.g. by suggesting c_i at random on some interval (a_i, b_i) or using some probability distribution $u_i(c_i)$.
3. Calculate $f_{set} = f(c_1, c_2, \ldots, c_N)$. If $(f_{set} < f_{min})$ then assign a new value for the minimum $f_{min} = f_{set}$ and store the corresponding set in the memory $\{c_i\}_{best} = \{c_i\}$. Continue from point **2.**

After many attempts, you will find the smallest value of f when points are selected at random in wide range. Next, you may refine your search by making the selection of sets $\{c_i\}$ in the close vicinity of point $\{c_i\}_{best}$. Then refine more if not satisfied, etc. There is no guarantee that you will find the global minimum if the number of attempts is limited, but you have very good chances to get it if the global minimum valley is relatively wide.

Problem. Find the minimum of the function $f = 1.2 - 3.3c_1 + 2c_1^2 + 2c_2 - c_2^2 + 4c_2^3 + 0.5c_2^4 - 1.5c_3 + 0.44c_3^2$ **where** $c_i \in (-\infty, \infty)$ **using random numbers with accuracy up to 5 meaningful digits.**

The variational principle for the search of the ground state may be formulated as follows. Let $|\,\alpha\rangle$ is the set of eigenvalue states of the Hamiltonian H, i.e. $H|\,\alpha\rangle = E_\alpha|\,\alpha\rangle$. Since the set is complete, we write an arbitrary state ψ as

$$\psi = \sum_\alpha p_\alpha|\,\alpha\rangle\ , \quad \sum_\alpha |p_\alpha|^2 = 1\ ,$$

and use it to calculate the average energy

$$\langle E\rangle_\psi = \frac{\langle\,\psi|\,H|\,\psi\rangle}{\langle\,\psi|\,\psi\rangle} = \frac{\sum_\alpha |p_\alpha|^2 E_\alpha}{\sum_\alpha |p_\alpha|^2}\,. \equiv E_G + \frac{\sum_\alpha |p_\alpha|^2 (E_\alpha - E_G)}{\sum_\alpha |p_\alpha|^2} \geq E_G\,.$$

Unless all coefficients except p_G are zero the average energy of the state ψ is larger than E_G. If we use a set of parameters $\{c_i\}$ to write down the trial wavefunction ψ, then optimization procedures looking for the best minimum of function $f(c_1, c_2, \ldots, c_K) = \langle E\rangle_\psi$ can be used to determine the best approximation to the ground state within a given parameterization $\psi[\{c_i\}; R)]$, where, as usual, $R = \{r_j\}$ denotes all particle coordinates.

Easy to say what the f-function is, but hard to get it if the number of particles is enormous. If we write

$$f = \langle E\rangle_\psi \equiv \frac{\int dR \psi^*(R) H\psi(R)}{\int dR \psi^*(R)\psi(R)} \equiv \frac{\int dR |\psi(R)|^2 \frac{H\psi(R)}{\psi(R)}}{\int dR |\psi(R)|^2}\,,$$

then we recover a familiar Monte Carlo setup: just change notations

$$\nu = R\,, \quad W_\nu = |\psi(R)|^2\,, \quad \text{and} \;\; A_\nu = \frac{H\psi(R)}{\psi(R)}\,,$$

It means that the value of f should itself be found from the MC simulation! It can be done using standard rules—generate R configurations according to the probability density $|\psi(R)|^2$ (e.g. by making local shifts of one of the coordinates $(\ldots, r_i, \ldots) \to (\ldots, r_i' = r_i + \Delta r_i, \ldots)$, and accepting them with the acceptance ratio $|\psi(R')|^2/|\psi(R)|^2$) and collect statistics of $H\psi(R)/\psi(R)$. Once the value of f is determined with some reasonable accuracy we use it as an input for the optimization procedure. This is the essence of the variational MC approach.

There is another way to form a function f which has the smallest possible value for the system eigenvalue state. Note, that for the eigenvalue state (and *only* for the eigenvalue state) the quantity A, also called a **local energy**, is R independent! It means that the variance

$$\tilde{f} = \langle\,(A - \langle A\rangle_\psi)^2\,\rangle_\psi = \langle\,A^2\,\rangle_\psi - f^2\,,$$

is a measure of how well ψ is approximating an eigenstate. At this point one may choose to optimize $\tilde{f}$ instead of f. Since any eigenstate is a minimum of $\tilde{f}$ with $\tilde{f}_{min} = 0$, one still has to check which $\tilde{f}$ minimum has the lowest

energy f. It is not a problem since the calculation $\tilde{f}$ necessarily involves the calculation of f.

The most important part of the variational MC approach is a good choice of the trial wavefunction. In many cases other approximate numerical schemes and experimental observations are used to motivate the choice and range of parameters for ψ. It is all about better convergence—if you have an infinite time to run the code and an exponential number of parameters to optimize (in practice, the $\{c_i\}$ sets do not involve more than several hundred parameters) then almost any guess will do the job. For example, in simulations of ^4He a popular choice is the Jastrow wavefunction

$$\psi = \exp\left\{ -\sum_{i<j}^{N} g_2(r_i - r_j) \right\} ,$$

which is the product of pairwise correlations between particles. One may use several parameters to characterize the $g_2(r)$ function. If not happy, add terms which involve three-particle correlations, $g_3(r_i - r_j, r_i - r_k)$, four-particle correlations, g_4, etc.

Other methods: Stochastic Series Expansion (SSE) and Diffusion MC (DMC)

There many other quantum Monte Carlo methods such as Stochastic Series Expansion method, Diffusion Monte Carlo (with or without the fixed-node approach), cluster methods, determinant MC, etc. We have no time to review them all in detail, but I will try to give you an idea of what some of them do (I will cover only SSE and DMC methods below).

The **Stochastic Series Expansion (SSE)** method starts from conventional decomposition (using previously introduced notations)

$$H = K + U \ , \quad U|\,\alpha\rangle = U_\alpha|\,\alpha\rangle \ , \quad \langle\,\alpha|\,K|\,\alpha\rangle = 0 \ ,$$

$$Z = \sum_\alpha \langle\,\alpha|\,e^{-\beta H}|\,\alpha\rangle \ .$$

The next step is simple in principle (Handscomb, 1962), but took a lot of time to figure out how to manage it efficiently in simulations (Sandvik 1991-97) [moreover, it was the first *exact*, i.e. no systematic errors, QMC method]

$$Z = \sum_\alpha \langle\,\alpha|\,\sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} H^n|\,\alpha\rangle = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_\alpha \langle\,\alpha|\,(K+U)^n|\,\alpha\rangle \ . \ .$$

The $(K+U)^n$ expression can be written as a sum over all possible strings of n operators

$$(K+U)^n = \sum_{\sigma_1=0}^{1} \ldots \sum_{\sigma_n=0}^{1} O_{\sigma_1} O_{\sigma_2} \ldots O_{\sigma_n} \equiv \sum_{\{\sigma_i\}} O_{\sigma_1} \ldots O_{\sigma_n} \ ,$$

where $O_0 = U$ and $O_1 = K$. Thus

$$Z = \sum_{n=0}^{\infty} \sum_{\{\sigma_i\}} \frac{(-\beta)^n}{n!} \sum_\alpha \langle\,\alpha|\,O_{\sigma_1} \ldots O_{\sigma_n}|\,\alpha\rangle \ .$$

$$= \sum_{n=0}^{\infty} \sum_{\{\sigma_i\}} \sum_{\alpha_1,\alpha_2 \ldots \alpha_n} \left(\frac{(-\beta)^n}{n!} (O_{\sigma_1})_{\alpha_1\alpha_2} (O_{\sigma_2})_{\alpha_2\alpha_3} \ldots (O_{\sigma_n})_{\alpha_n\alpha_1} \right) \ .$$

At this point we notice that the sequence of basis states α_i unambigously determines the sequence of O_{σ_i} operators because if $\alpha_i = \alpha_{i+1}$ then only

U has a nonzero matrix element, if $\alpha_i \neq \alpha_{i+1}$ then only K has a nonzero matrix element. Thus, we may drop the summation over $\{\sigma_i\}$ as redundant. Furthermore, both U and K are sums of many terms. Since U is diagonal, it is not a problem to deal with all of them at once, e.g. if $\mid \alpha\rangle$ is a collection of occupation numbers on lattice sites, $\mid \alpha\rangle = \mid n_1, n_2, \ldots\rangle$, then $U_\alpha = \sum_{ij} V_{ij} n_i n_j - c \sum_i n_i$. The non-diagonal terms typically, i.e. in most models, change only one or two indeces in the many body set $\{n_1, n_2, \ldots n_N\}$, and it is clear from the structure of α_i and α_{i+1} which term in K was responsible for the transition. For example, if the two states are different only by the value of the occupation number on sites j and j' which have changed from $n_j, n_{j'}$ to $n_j + 1, n_{j'} - 1$ in the bosonic system with $K = -\Delta_0 \sum_{<kk'>} b^\dagger_{k'} b_k$ then we know that the term $-\Delta_0 b^\dagger_j b_{j'}$ was responsible for the transition and its matrix element is $-\Delta_0 \sqrt{(n_j + 1) n_{j'}}$. It means that for any sequence of $\{\alpha_i\}$ one can easily calculate the corresponding matrix elements. Putting everything together

$$Z = \sum_{n=0}^{\infty} \sum_{\{\alpha_i\}} W_\nu \ ,$$

where

$$W_{\nu=n,\{\alpha_i\}} = \frac{(-\beta)^n}{n!} H_{\alpha_1 \alpha_2} \ldots H_{\alpha_n \alpha_1} \ .$$

This expression is obvious from the very beginning; the purpose of the decomposition $H = K + U$ was to show that diagonal and non-diagonal elements are treated differently.

In a way, one may consider the sequence of states in the string $\alpha_1, \alpha_2, \ldots, \alpha_n$ as a "trajectory" evolving from left to the right, though there is no true time variable in this method. This analogy turns out to be very useful in understanding of how to handle the data structure efficiently. The stochastic series expansion method samples α-"trajectories" of different length n, or, due to the unambiguous relation between the α-"trajectory" and the operator string in the expression H^n, all possible operator sequencies originating from H^n. I will stop short on the algorithmic details, and simply note that the most efficient way of updating operator strings is based on adding and moving "worms"=($b^\dagger_{j_1} = masha$ and $b_{j_2} = ira$ operators) around.

The **Diffusion Monte Carlo (DMC)** method closely resembles the solution of kinetic equiations using random walkers (Meyers, 1956; Kalos,

1974; Ceperley, 1980-86). Indeed, Schrödinger Eq. in imaginary time

$$\frac{\partial\Psi(\tau,R)}{\partial\tau} = (E-H)\Psi(\tau,R) = D\nabla^2\Psi(\tau,R) + [E-V(R)]\Psi(\tau,R)$$

where $D = 1/2m$ and E is some constant, is identical to the equation describing a random walker in the $3N$-dimensional dangerous wonderland which has the property of duplicating the walker with the rate $\Gamma = E - V(R)$ if $\Gamma > 0$ or making him die with the rate $V(R) - E$ if $\Gamma < 0$. Obviously, the number of walkers is not conserved in this problem, and in the long run their number may grow to infinity or shink to zero. To understand properties of the solution, we use the egenvalue set of H (see paragraph on Variational MC) to write identically

$$\Psi(0,R) = \sum_\alpha p_\alpha |\,\alpha\rangle \ ,$$

$$\Psi(\tau,R) = \sum_\alpha e^{\tau(E-H)} p_\alpha |\,\alpha\rangle = \sum_\alpha e^{\tau(E-E_\alpha)} p_\alpha |\,\alpha\rangle \ .$$

In the limit $\tau \to \infty$ all terms with $E - E_\alpha < 0$ will decay to zero and the ground state will be dominating in $\Psi(\tau,R)$ because it has the smallest decay rate or the largest inflation rate. If E, which is arbitrary so far, is smaller than E_G then all terms decay to zero. This is unfortunate: though $|\,G\rangle$ is the largest term in the sum it has a vanishing amplitude. If $E > E_G$ the solution goes to infinity and may be unstable numerically. The best choice is then to have $E = E_G$—this will correspond to all terms decaying to zero except the ground state which remains finite.

The MC procedure is then to perform conventional random walks in the coordinate space R using many (say several thousand) walkers with additional feature that the number of walkers may increase/decrease with the rate $E - V(R)$ depending on the point in space. The value of E must be adjusted so that in the long run the total number of random walkers is stable and nearly constant in time. Once this quasistatic regime has been reached, one may collect space histograms $hist(R)$ counting the number of visits to R-bins. This procedure should give us the ground state energy and the ground state wavefunction.

There is one useful trick to improve the performance of the method. Let $\phi(R)$ is the best we can do to approximate the ground state, e.g. it can be the result of the variational MC method. We write $\psi(\tau,R) = f(\tau,R)/\phi(R)$ and substitute this into the Schrödinger Eq.

$$\dot{f}/\phi = (D\nabla^2 f)/\phi + 2D(\nabla(1/\phi))(\nabla f) + D\nabla^2(1/\phi)f + [E-V]f/\phi \ ,$$

or

$$\dot{f}(\tau,R) = D\nabla\Big(\nabla f(\tau,R) - F(R)f(\tau,R)\Big) + \Big[E - A(R)\Big]f(\tau,R) \ ,$$

where the "drift force" is

$$F(R) = \nabla \ln \phi^2(R) \ ,$$

and A is the local energy introduced previously

$$A(R) \ = \ \frac{-D\nabla^2\phi(R) + V\phi(R)}{\phi(R)} \ = \ \frac{H\phi(R)}{\phi(R)} \ = \ \frac{\phi^*(R)H\phi(R)}{\phi^*(R)\phi(R)} \ .$$

The idea is to minimize local fluctuations of the increase/decrease rates and make the whole scheme as close as possible to the conventional random walk in external field, i.e. we improve on top of the known approximation $\phi(R)$. This technique is nothing but another "face" of the importance sampling, or reweighing method in MC.

Random walks in the presence of external field are performed as updates (in time $\Delta\tau$)

$$R_{\tau+\Delta\tau} = R(\tau) + \xi + DF(R)\Delta\tau \ ,$$

where $\xi = 9\xi_1, \ldots, \xi_N)$ is the Gaussian random variable distributed according to the diffusion propagator

$$\prod_{i=1}^{N} \Big(4\pi D\Delta\tau\Big)^{-3/2} e^{-\xi_i^2/2D\Delta\tau} \ .$$

This formula derives from the solution of the diffusion Eq. in Fourier space (with constant force F)

$$G(\tau,k) = e^{-Dk^2\tau - ikDF\tau} \longrightarrow$$

$$G(\tau,R) = \int dk e^{ikR} G(\tau,k) = \int dk e^{ik(R-DF\tau)} e^{-Dk^2\tau} = G^{(0)}(\tau, R - DF\tau) \ ,$$

where $G^{(0)}$ is the diffusion propagator in the absence of the force.