Solving Square Lattice Hubbard Model using DQMC

Xiao Yan Xu
xyxu@iphy.ac.cn
Institute of Physics, Chinese Academy of Sciences
100190 Beijing, China

August 5, 2016

1 DQMC code implement

1.1 Model and partition function

The half-filling Hubbard model on square lattice can be written as

$$\hat{H} = -t \sum_{\langle ij \rangle \sigma} \hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + h.c. + U \sum_{i} (\hat{n}_{i\uparrow} - \frac{1}{2})(\hat{n}_{i\downarrow} - \frac{1}{2})$$

To perform the DQMC simulation of this model, we start with representing the partition function as a sum over a configuration space. The partition function writes

$$\begin{aligned} Z &= \mathrm{Tr}\left[e^{-\beta \hat{H}}\right] \\ &= \mathrm{Tr}\left[\left(e^{-\Delta\tau \hat{H}}\right)^{M}\right] \end{aligned}$$

First we do trotter decomposition

$$e^{-\Delta\tau \hat{H}} \approx e^{-\Delta\tau \hat{H}_I} e^{-\Delta\tau \hat{T}}$$

To take care of the interaction part, we need do the HS transformation

$$\begin{aligned} e^{-\Delta\tau \hat{H}_I} &= \prod_i e^{-\Delta\tau U(\hat{n}_{i\uparrow} - \frac{1}{2})(\hat{n}_{i\downarrow} - \frac{1}{2})} \\ &= \prod_i \lambda \sum_{s_{i,\tau} = \pm 1} e^{\alpha s_{i,\tau}(\hat{n}_{i\uparrow} - \hat{n}_{i\downarrow})} \\ &= \lambda^N \sum_{s_{i,\tau} = \pm 1} \left(\prod_i e^{\alpha s_{i,\tau} \hat{n}_{i\uparrow}} \prod_i e^{-\alpha s_{i,\tau} \hat{n}_{i\downarrow}} \right) \end{aligned}$$

To get more efficient code performance, we also do the check-board decomposition of kinetic part. We can classify all hopping bonds as two families which both are combined of disconnected plaques. Within each family, all plaques are commutative, thus we can further write down the summation over plaques in the exponential as products.

$$\begin{aligned} e^{-\Delta\tau \hat{T}} &= \prod_{\sigma} e^{-\Delta\tau t \sum_{\langle ij \rangle} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right)} \\ &= \prod_{\sigma} e^{-\Delta\tau t \left[\sum_{\langle ij \rangle \in \mathrm{fam}_1} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right) + \sum_{\langle ij \rangle \in \mathrm{fam}_2} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right)\right]} \\ &\approx \prod_{\sigma} e^{-\Delta\tau t \left[\sum_{\langle ij \rangle \in \mathrm{fam}_1} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right)\right]} e^{-\Delta\tau t \left[\sum_{\langle ij \rangle \in \mathrm{fam}_2} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right)\right]} \\ &= \prod_{\sigma} \prod_{m=1}^{2} \prod_{n=1}^{N/4} e^{-\Delta\tau t \sum_{\langle ij \rangle \in \square_{mn}} \left(\hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma} \hat{c}_{i\sigma}\right)} \end{aligned}$$

The second last step is also a trotter decomposition. Write the interaction and kinetic part together, we have

$$\begin{aligned} e^{-\Delta\tau\hat{H}} &= \lambda^N \sum_{s_{i,\tau}=\pm 1} \\ &\left(\prod_i e^{\alpha s_{i,\tau}\hat{n}_{i\uparrow}}\right)\left(\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\sum_{\langle ij\rangle\in\square_{mn}}\left(\hat{c}^\dagger_{i\uparrow}\hat{c}_{j\uparrow}+\hat{c}^\dagger_{j\uparrow}\hat{c}_{i\uparrow}\right)}\right) \\ &\left(\prod_i e^{-\alpha s_{i,\tau}\hat{n}_{i\downarrow}}\right)\left(\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\sum_{\langle ij\rangle\in\square_{mn}}\left(\hat{c}^\dagger_{i\downarrow}\hat{c}_{j\downarrow}+\hat{c}^\dagger_{j\downarrow}\hat{c}_{i\downarrow}\right)}\right) \end{aligned}$$

With those tricks, and after we trace over the fermion degrees of freedom, we get the following formulation for partition function

$$Z = \lambda^{NM}\sum_{\{s_{i,\tau}\}}\prod_\sigma \det\left[\mathbf{1}+\mathbf{B}^\sigma_M\mathbf{B}^\sigma_{M-1}\cdots\mathbf{B}^\sigma_2\mathbf{B}^\sigma_1\right]$$

with

$$\mathbf{B}^\uparrow_\tau = e^{\alpha\mathbf{Diag}(\vec{S}_\tau)}\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}$$

$$\mathbf{B}^\downarrow_\tau = e^{-\alpha\mathbf{Diag}(\vec{S}_\tau)}\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}$$

Here we directly see the effect of check-board decomposition, namely, decomposing the huge and dense kinetic matrix to products of many small (sparse) matrices. Note that our configuration space are all possible values of auxiliary fields $\{s_{i,\tau}\}$ on a "space-time" lattice with size $L\times L\times M$.

1.2 Update strategy

We will use local update scheme to update the auxiliary fields on the "space-time" lattice. The strategy comes like this:

we start with an initial random configuration, try to flip auxiliary fields one by one at every sites of time slice plane from $\tau = M$ to $\tau = 1$ and then back and forth.

Following, we will use short notations

$$\mathbf{B}^\sigma(\beta,\tau)\equiv\mathbf{B}^\sigma_M\cdots\mathbf{B}^\sigma_{\tau+1}$$

$$\mathbf{B}^\sigma(\tau,0)\equiv\mathbf{B}^\sigma_\tau\cdots\mathbf{B}^\sigma_1$$

The accept ratio of local update is

$$\begin{aligned} \mathcal{R} &= \prod_\sigma \frac{\det\left(\mathbf{1}+\mathbf{B}^\sigma(\beta,\tau)(\mathbf{1}+\mathbf{\Delta}^\sigma(i,\tau))\mathbf{B}^\sigma(\tau,0)\right)}{\det\left(\mathbf{1}+\mathbf{B}^\sigma(\beta,0)\right)} \\ &= \prod_\sigma \det\left(\mathbf{1}+(\mathbf{1}+\mathbf{B}^\sigma(\beta,0))^{-1}\mathbf{B}^\sigma(\beta,\tau)\mathbf{\Delta}^\sigma(i,\tau)\mathbf{B}^\sigma(\tau,0)\right) \\ &= \prod_\sigma \det\left(\mathbf{1}+(\mathbf{1}+\mathbf{\Delta}^\sigma(i,\tau)\mathbf{B}^\sigma(\tau,0)\mathbf{B}^\sigma(\beta,0))^{-1}\mathbf{B}^\sigma(\beta,\tau)\right) \\ &= \prod_\sigma \det\left(\mathbf{1}+\mathbf{\Delta}^\sigma(i,\tau)(1-\mathbf{G}^\sigma(\tau,\tau))\right) \\ &= \prod_\sigma \left(1+\Delta^\sigma_{ii}(i,\tau)\left(1-G^\sigma_{ii}(\tau,\tau)\right)\right) \end{aligned}$$

and the equal time Green function updates as

$$\mathbf{G}^\sigma(\tau,\tau)\to\mathbf{G}^\sigma(\tau,\tau)-\frac{1}{\mathcal{R}^\sigma}\mathbf{G}^\sigma(\tau,\tau)\mathbf{\Delta}^\sigma(i,\tau)(\mathbf{1}-\mathbf{G}^\sigma(\tau,\tau))$$

1.3 Propagating Green function

The update strategy tells us the equal time Green function is the fundamental quantity in this algorithm. Using the local update scheme, we can easily update Green function when we flip auxiliary fields of sites in the same time slice plane. Then how to calculate equal time Green function at next time, namely, $\mathbf{G}^\sigma(\tau-1,\tau-1)$ or $\mathbf{G}^\sigma(\tau+1,\tau+1)$. Here we will give an example for calculating $\mathbf{G}^\sigma(\tau+1,\tau+1)$ when sweep from $\tau=1$ to $\tau=M$. The following four steps "mmthr, mmthlm1, mmuur, mmuulm1" are just names of four subroutines which do the following matrix operations

mmthr:

$$\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\mathbf{G}^\sigma(\tau,\tau)$$

mmthlm1:

$$\begin{aligned}&\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\mathbf{G}^\sigma(\tau,\tau)\left(\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\right)^{-1}\\ =&\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\mathbf{G}^\sigma(\tau,\tau)\prod_{m=2}^{1}\prod_{n=1}^{N/4} e^{\Delta\tau t\mathbf{T}_{\square_{mn}}}\end{aligned}$$

mmuur:

$$e^{\sigma\alpha\mathbf{Diag}(\vec{S}_{\tau+d\tau})}\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\mathbf{G}^\sigma(\tau,\tau)\prod_{m=2}^{1}\prod_{n=1}^{N/4} e^{\Delta\tau t\mathbf{T}_{\square_{mn}}}$$

mmuulm1:

$$e^{\sigma\mathbf{Diag}(\vec{S}_{\tau+d\tau})}\prod_{m=1}^{2}\prod_{n=1}^{N/4} e^{-\Delta\tau t\mathbf{T}_{\square_{mn}}}\mathbf{G}^\sigma(\tau,\tau)\prod_{m=2}^{1}\prod_{n=1}^{N/4} e^{\Delta\tau t\mathbf{T}_{\square_{mn}}}e^{-\sigma\alpha\mathbf{Diag}(\vec{S}_{\tau+d\tau})}$$

The total effect of above four steps can be written in a short formula

$$\mathbf{G}^\sigma(\tau+1,\tau+1) \;=\; \mathbf{B}^\sigma(\tau+1,\tau)\mathbf{G}^\sigma(\tau,\tau)\mathbf{B}^\sigma(\tau+1,\tau)^{-1}$$

which propagates the Green function from τ to $\tau+1$. The sweep from $\tau=M$ to $\tau=1$ is very similar.

1.4 Numerical stabilization

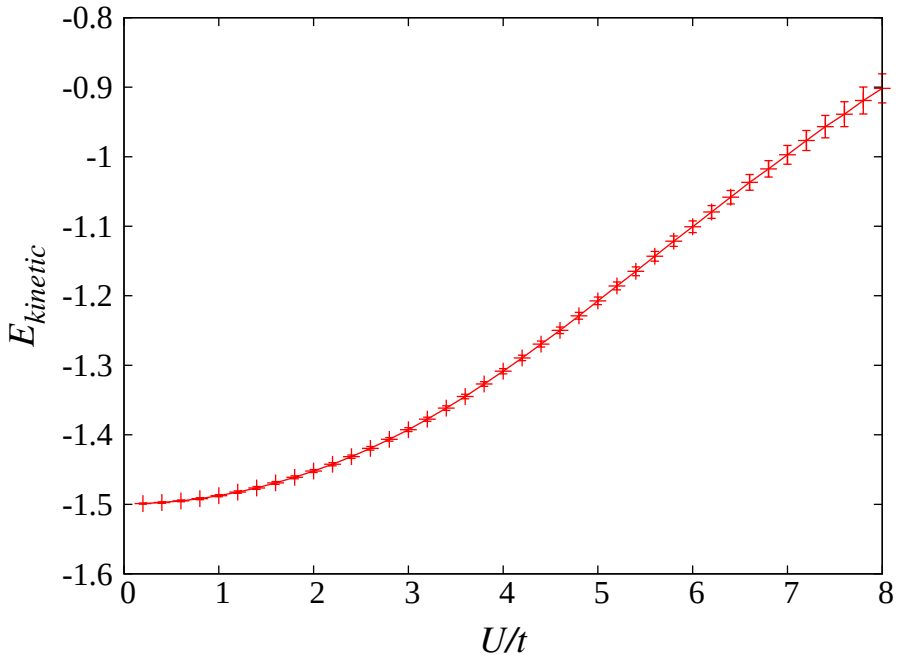
The Green function propagating process will accumulate numerical errors. We need to do numerical stabilization after several steps of propagating. Here is the stabilization scheme for equal time Green function. We recalculate the equal time Green function after several steps ("nwrap" in code) of propagating using following equation.

$$\begin{aligned}\mathbf{G}(\tau,\tau) &= [\mathbf{1}+\mathbf{B}(\tau,0)\mathbf{B}(\beta,\tau)]^{-1}\\ &= [\mathbf{1}+\mathbf{U}_R\mathbf{D}_R\mathbf{V}_R\mathbf{V}_L\mathbf{D}_L\mathbf{U}_L]^{-1}\\ &= \mathbf{U}_L^{-1}\left[(\mathbf{U}_L\mathbf{U}_R)^{-1}+\mathbf{D}_R\left(\mathbf{V}_R\mathbf{V}_L\right)\mathbf{D}_L\right]^{-1}\mathbf{U}_R^{-1}\\ &= \mathbf{U}_L^{-1}\left[(\mathbf{U}_L\mathbf{U}_R)^{-1}+\mathbf{D}_R^{\max}\mathbf{D}_R^{\min}\left(\mathbf{V}_R\mathbf{V}_L\right)\mathbf{D}_L^{\min}\mathbf{D}_L^{\max}\right]^{-1}\mathbf{U}_R^{-1}\\ &= \mathbf{U}_L^{-1}\left(\mathbf{D}_L^{\max}\right)^{-1}\left[\left(\mathbf{D}_R^{\max}\right)^{-1}\left(\mathbf{U}_L\mathbf{U}_R\right)^{-1}\left(\mathbf{D}_L^{\max}\right)^{-1}+\mathbf{D}_R^{\min}\mathbf{V}_R\mathbf{V}_L\mathbf{D}_L^{\min}\right]^{-1}\left(\mathbf{D}_R^{\max}\right)^{-1}\mathbf{U}_R^{-1}\end{aligned}$$

2 Results

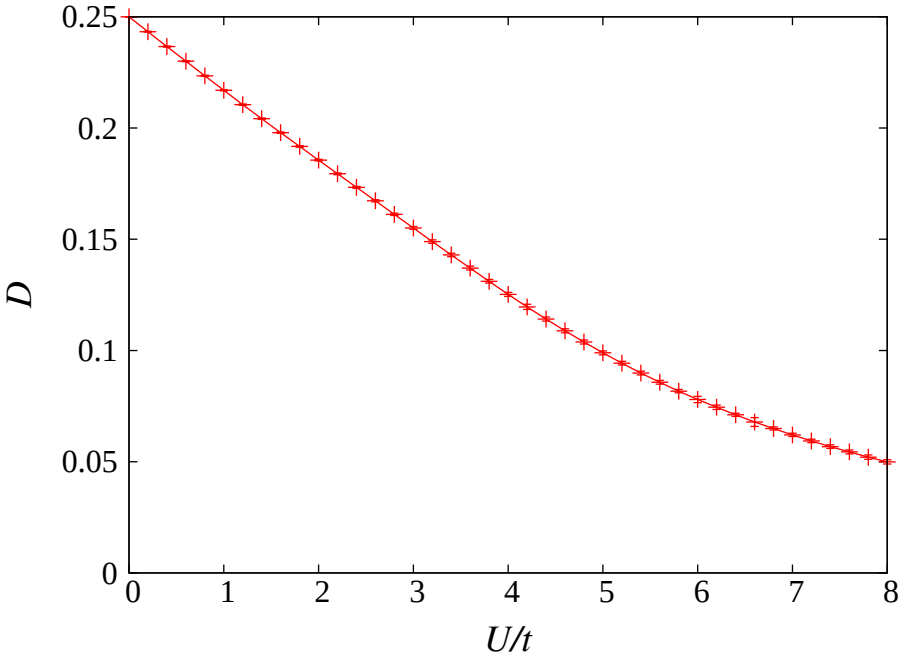
2.1 Kinetic energy

Here is the kinetic energy varying with interaction U. The calculation parameters is $L=4$, $\beta=4$.



2.2 Double occupancy

We can define double occupancy $D = \langle n_{i\uparrow} n_{i\downarrow} \rangle$ as the order parameter for Mott transition. Here we show the results with parameters $L = 4$, $\beta = 4$ and U varying from $0.0t$ to $8.0t$.

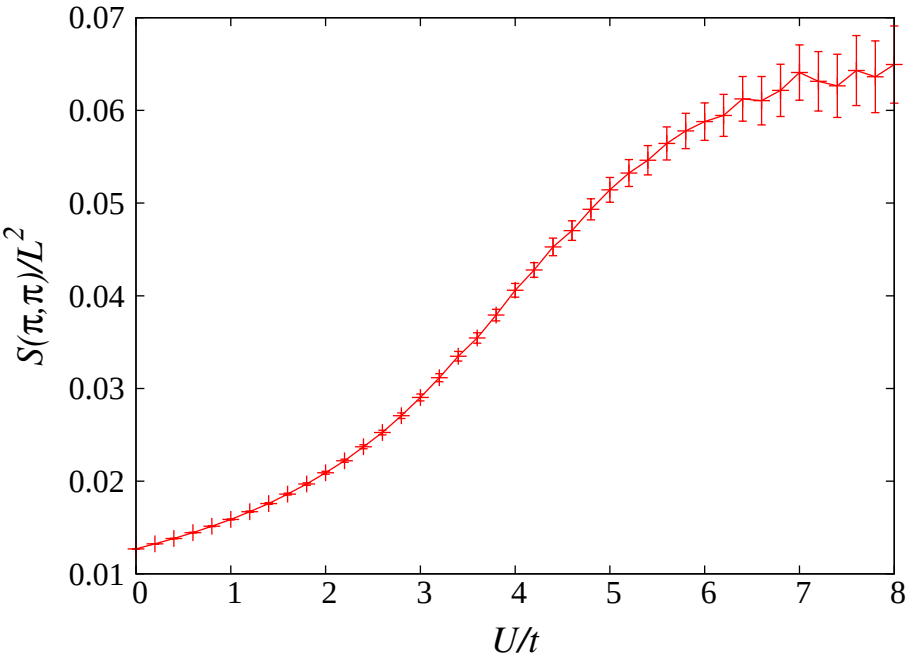


2.3 Magnetization

We measure the z-component antiferromagnetic structure factor $S(\pi, \pi)$, which is defined as

$$S(\mathbf{Q}) = \frac{1}{L^2} \sum_{ij} e^{-i\mathbf{Q}\cdot(\mathbf{r}_i - \mathbf{r}_j)} \langle \hat{s}_i^z \hat{s}_j^z \rangle$$

with $\mathbf{Q} = (\pi, \pi)$. The relation between magnetization m_s and $S(\pi, \pi)$ is $m_s^2 = 3S(\pi, \pi)/L^2$, where factor 3 counts three components of $SU(2)$ spin. Here we show $S(\pi, \pi)/L^2$ varying with interaction with parameters $L = 4$, $\beta = 4$.



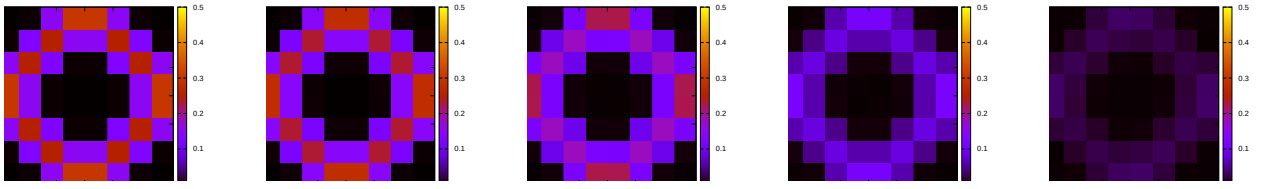
We know that at large U limit, the half-filling Hubbard model on square lattice will reduce to a Heisenberg model. The magnetization of Heisenberg model is about 0.3, which can be calculated from spin wave theory and also confirmed in experiments. Therefore, in the large U limit and in the thermodynamic limit, the square of magnetization will approach 0.09.

2.4 Spectral function

Real frequency information can be extracted from imaginary-time data using the identity

$$G(\mathbf{k}, \tau > 0) = \int_{-\infty}^{\infty} d\omega \frac{e^{-\omega(\tau-\beta/2)}}{2\cosh(\beta\omega/2)} A(\mathbf{k}, \omega)$$

where $A(\mathbf{k}, \omega)$ is the fermion spectral function. The $\cosh^{-1}$ term implies that this integral is sharply peaked around $\omega = 0$. We can approximate spectral function around Fermi level with $\beta G(\mathbf{k}, \beta/2)$. Following is the evolution of $G(\mathbf{k}, \beta/2)$ when increasing interaction and keep other parameters fixed ($L = 8$, $\beta = 8$).



$U = 0.0t$ $U = 1.0t$ $U = 2.0t$ $U = 3.0t$ $U = 4.0t$

$G(\mathbf{k}, \beta/2)$ gradually decreases when increasing interaction.