Documentation for the General QMC code

Author: Martin Bercx

November 5, 2016

Contents

1 Definition of the model Hamiltonian **2**
1.1 Formulation of the QMC 2
1.2 Structure of the hopping matrix $\mathbf{T}$ and the interaction matrices $\mathbf{V}^{(k)}$ 3
1.3 The Hubbard-Stratonovich decomposition 4

2 Implementation of a model Hamiltonian **4**
2.1 The `Operator` variable 4
2.2 The observables 5
2.3 The lattice 5

3 Input and output files **5**

4 Walkthrough: the $SU(2)$-Hubbard model on a square lattice **5**
4.1 Hopping term 5
4.2 Interaction term 6
4.3 Definition of the square lattice 6
4.4 Observables for the Hubbard model 6

5 Tutorial: set up a model Hamiltonian **7**

1 Definition of the model Hamiltonian

Notation: Hats for second quantized operators, bold for matrices.
Structure:
1) We first want to define the model.
2) implementation of the QMC.
3) Data structure
4) Practical implementation and some simple test cases.

The class of solvable models includes Hamiltonians $\mathcal{H}$ having the following general form: $\mathcal{H} = \mathcal{H}_T + \mathcal{H}_V + \mathcal{H}_I + \mathcal{H}_{0,I}$, where

$$\mathcal{H}_T = \sum_{k=1}^{M_T} \sum_{s=1}^{N_{fl}} \sum_{\sigma=1}^{N_{col}} \sum_{x,y} c^\dagger_{x\sigma s} T^{(ks)}_{xy} c_{y\sigma s} \equiv \sum_{k=1}^{M_T} \hat{T}^{(k)} \tag{1}$$

$$\mathcal{H}_V = -\sum_{k=1}^{M_V} U_k \left\{ \sum_{s=1}^{N_{fl}} \sum_{\sigma=1}^{N_{col}} \left[\left(\sum_{x,y} c^\dagger_{x\sigma s} V^{(ks)}_{xy} c_{y\sigma s} \right) - \alpha_{ks} \right] \right\}^2 \equiv -\sum_{k=1}^{M_V} U_k \left(\hat{V}(k) \right)^2 \tag{2}$$

$$\mathcal{H}_I = \sum_{k=1}^{M_I} \hat{\sigma}^z_k \left\{ \sum_{s=1}^{N_{fl}} \sum_{\sigma=1}^{N_{col}} \left[\sum_{x,y} c^\dagger_{x\sigma s} I^{(ks)}_{xy} c_{y\sigma s} \right] \right\} \equiv \sum_{k=1}^{M_I} \hat{\sigma}^z_k \hat{I}^{(k)} \ . \tag{3}$$

The indices have the following meaning:

- The number of fermion *flavors* is set by N_{fl}. After the Hubbard Stratonovitch transformation, the action will be block diagonal in the flavor index.
- The number of fermion *colors* is set by N_{col}. The Hamiltonian is invariant under SU(N_{col}) rotations. Note that In the code $N_{col} \equiv NSUN$.
- The indices x, y label lattice sites where $x, y = 1, \cdots N_{dim}$. N_{dim} is the total number of spacial vertices: $N_{dim} = N_{unit\ cell} N_{orbital}$, where $N_{unit\ cell}$ is the number of unit cells of the underlying Bravais lattice and $N_{orbital}$ is the number of (spacial) orbitals per unit cell Check the definition of $N_{orbital}$ in the code.
- Therefore, the matrices $\boldsymbol{T}^{(ks)}$, $\boldsymbol{V}^{(ks)}$ and $\boldsymbol{I}^{(ks)}$ are of dimension $N_{dim} \times N_{dim}$
- The number of interaction terms is labelled by M_V and M_I. $M_T > 1$ would allow for a checkerboard decomposition.
- $\hat{\sigma}^z_k$ is an Ising variable which couples to a general one-body term.
- $\mathcal{H}_{0,I}$ gives the dynamics of the Ising variable.

Note that the matrices $\boldsymbol{T}^{(s)}$ and $\boldsymbol{V}^{(ks)}$ explicitly depend on the flavor index s but not on the color index σ. The color index σ only appears only in the second quantized operators such that the Hamiltonian is manifestly SU(N_{col}) symmetric.

1.1 Formulation of the QMC

The formulation of the Monte Carlo simulation is based on the following.

- We will work in a basis where $\hat{\sigma}^z_k$ is diagonal.
- We will discretize the imaginary time propagation: $\beta = \Delta\tau L_{\text{Trotter}}$

- We will the discrete Hubbard Stratonovitch transformation:

$$e^{\Delta\tau\lambda A^2} = \sum_{l=\pm1,\pm2} \gamma(l) e^{\sqrt{\Delta\tau\lambda}\eta(l)O} + \mathcal{O}(\Delta\tau^4) \tag{4}$$

  where the fields η and γ take the values:

$$\gamma(\pm1) = 1 + \sqrt{6}/3, \quad \gamma(\pm2) = 1 - \sqrt{6}/3$$
$$\eta(\pm1) = \pm\sqrt{2\left(3 - \sqrt{6}\right)}, \quad \eta(\pm2) = \pm\sqrt{2\left(3 + \sqrt{6}\right)}.$$

- From the above it follows that the Monte Carlo configuration space is given by:

$$C = \{s_{i,\tau}, l_{j,\tau} \text{ with } i = 1 \cdots M_I, j = 1 \cdots M_V, \tau = 1, L_{Trotter}\} \tag{5}$$

With the above, the partition function of the model can be written as follows.

$$Z = \mathrm{Tr} e^{-\beta\mathcal{H}} = \mathrm{Tr}\left[e^{-\Delta\tau\mathcal{H}_{0,I}} \prod_{k=1}^{M_T} e^{-\Delta\tau\hat{T}^{(k)}} \prod_{k=1}^{M_V} e^{\Delta\tau U_k \left(\hat{V}^{(k)}\right)^2} \prod_{k=1}^{M_I} e^{-\Delta\tau\hat{\sigma}_k\hat{I}^{(k)}}\right]^{L_{\mathrm{Trotter}}}$$
$$\sum_C \left(\prod_{j=1}^{M_V}\prod_{\tau=1}^{L_{Trotter}} \gamma_{j,\tau}\right) e^{-S_{0,I}(\{s_{i,\tau}\})} \mathrm{Tr}_F \prod_{\tau=1}^{L_{Trotter}} \left[\prod_{k=1}^{M_T} e^{-\Delta\tau\hat{T}^{(k)}} \prod_{k=1}^{M_V} e^{\sqrt{\Delta\tau U_k}\eta_{k,\tau}\hat{V}^{(k)}} \prod_{k=1}^{M_I} e^{-\Delta\tau s_{k,\tau}\hat{I}^{(k)}}\right]$$

In the above, Tr runs over the Ising spins as well as over the fermionic degrees of freedom, and Tr_F only over the fermionc Fock space. $S_{0,I}(\{s_{i,\tau}\})$ is the action corresponding to the Ising Hamiltonian, and is only dependent on the Ising spins so that it can be pulled out of the fermionic trace. At this point, and since for a given configuration C we are dealing with a free propagation, we can integrate out the fermions to obtain a determinant:

$$\mathrm{Tr}_F \prod_{\tau=1}^{L_{Trotter}} \left[\prod_{k=1}^{M_T} e^{-\Delta\tau\hat{T}^{(k)}} \prod_{k=1}^{M_V} e^{\sqrt{\Delta\tau U_k}\eta_{k,\tau}\hat{V}^{(k)}} \prod_{k=1}^{M_I} e^{-\Delta\tau s_{k,\tau}\hat{I}^{(k)}}\right] = \tag{6}$$

We will first simplify the notation as:

- in the coupling g in the `Operator` structure (see Sec. **??**).
- as normalization constant in the definition of observables (see Sec. **??**)
- as exponent in the calculation of the phase factor and the Monte Carlo update ratio.

1.2 Structure of the hopping matrix T and the interaction matrices $\mathbf{V}^{(k)}$

In general, the matrices $\mathbf{T}^{(s)}$ and $\mathbf{V}^{(ks)}$ are sparse matrices. This property is used to minimize computational cost and storage. In the following, we discuss the implementation of the single-particle matrix representation $\mathbf{V}^{(ks)}$ of the interaction operator. The same logic applies for the hopping matrix $\mathbf{T}^{(s)}$.

We denote a subset of N_{eff} (in the code, N_{eff} is called just N) degrees of freedom here: sites by the set $[z_1, \cdots z_{N_{eff}}]$ and define it to contain only vertices for which an interaction term is defined:

$$V^{(ks)}_{xy} \neq 0 \quad \text{only if} \quad x, y \in [z^{(ks)}_1, \cdots z^{(ks)}_{N^{(ks)}_{eff}}] \,. \tag{7}$$

We define the projection matrices $\mathbf{P}^{(ks)}_V$ of dimension $N^{(ks)}_{eff} \times N_{dim}$:

$$(P^{(ks)}_V)_{i,z} = \delta_{z^{(ks)}_i, z} \,, \tag{8}$$

where $i \in [1, \cdots N_{eff}^{(ks)}]$ and $z \in [1, \cdots N_{dim}]$. The matrix operator $\boldsymbol{P}_V^{(ks)}$ picks out the non-vanishing entries of $\boldsymbol{V}^{(ks)}$, which are contained in the rank-$N_{eff}^{(ks)}$ matrix $\boldsymbol{O}_V^{(ks)}$:

$$\boldsymbol{V}^{(ks)} = \boldsymbol{P}_V^{(ks)T} \boldsymbol{O}_V^{(ks)} \boldsymbol{P}_V^{(ks)} \ , \tag{9}$$

and

$$V_{xy}^{(ks)} = (P_V^{(ks)})_{ix} \left[O_V^{(ks)}\right]_{ij} (P_V^{(ks)})_{jy} = \sum_{i,j}^{N_{eff}^{(ks)}} \delta_{z_i^{(ks)},x} \left[O_V^{(ks)}\right]_{ij} \delta_{z_j^{(ks)},y} \ . \tag{10}$$

Comment that the P matrices have only one non-vanishing entry per column. To set the interaction part, we therefore have to specify the following:

- the matrix elements $\left[O_V^{(k)}\right]_{ij}$
- the set $[z_1^{(k)}, \cdots z_{N_{eff}^{(k)}}^{(k)}]$
- the interaction strenghts U_k
- the numbers α_k.

Be more specific here what really has to specified in the actual code. The same logic also applies to the implementation of the hopping interaction be more specific.

1.3 The Hubbard-Stratonovich decomposition

Consider a single-particle (in other words bilinear) operator O_i. One obtains an approximation to the evolution operator by the following series expansion [**?**]

$$e^{-\Delta\tau O_i^2} = \sum_{s=\pm1,\pm2} \gamma(s) e^{i\sqrt{\Delta\tau}\eta(s)O_i} + \mathcal{O}(\Delta\tau^4) \ , \tag{11}$$

with

$$\begin{aligned} \gamma(\pm1) = (1+\sqrt{6}/3)/4 \ , \ \gamma(\pm2) = (1-\sqrt{6}/3)/4 \ , \\ \eta(\pm1) = \pm\sqrt{2(3-\sqrt{6})} \ , \ \eta(\pm2) = \pm\sqrt{2(3+\sqrt{6})} \ . \end{aligned} \tag{12}$$

Eq. (10) can be easily proven by expanding its right hand side to eighth order in O_i. The transformation introduces therefore two Ising fields s per lattice site i, taking the values ±1 and ±2. same label as the flavor index

2 Implementation of a model Hamiltonian

To implement a Hamiltonian which belongs to the class of Hamiltonians defined by Eq. (1), the user only has to write/modify a single subroutine. A template is given by `Hamiltonian_template.f90`. Existing model subroutines are `Hamiltonian_Hubb.f90`.

2.1 The `Operator` variable

In the code implementation, we define a structure called `Operator`. This structure variable `Operator` bundles several components that are needed to define and use an operator matrix in the program. In Fortran a structure variable like this is called a derived type. The components it contains are:

- the projector $\boldsymbol{P}_V$,
- the matrix $\boldsymbol{O}_V$,
- the effective dimension N_{eff},
- and a couple of auxiliary matrices and scalars.

In general, we will not only have one structure variable `Operator`, but a whole array of these structures.

Name of variable in the code	Description
`Op_V%N`	effective dimension N_{eff}
`Op_V%O`	matrix $\mathbf{O}_V$
`Op_V%U`	matrix containing the eigenvectors of $\mathbf{O}_V$
`Op_V%E`	eigenvalues of $\mathbf{O}_V$
`Op_V%P`	projection matrix $\mathbf{P}_V$
`Op_V%N_non_zero`	number of non-vanishing eigenvalues of $\mathbf{O}_V$
`Op_V%g`	coupling strength in Hubbard-Stratonovich transformation
`Op_V%alpha`	constant
`Op_V%type`	integer parameter to set the type of Hubbard-Stratonovich transformation Possible Issue: type is also a Fortran keyword

Table 1: Components of the `Operator` structure variable `Op_V`.

2.2 The observables

2.3 The lattice

3 Input and output files

4 Walkthrough: the $SU(2)$-Hubbard model on a square lattice

In this section, we describe the subroutine `Hamiltonian_Hub.f90` which is an implementation of the Hubbard model on the square lattice. The $SU(2)$-symmetric Hubbard model is given by

$$\mathcal{H} = -t \sum_{\sigma=1}^{2} \sum_{\langle x,y \rangle} \left(c^\dagger_{x\sigma} c_{y\sigma} + \text{H.c.} \right) + \frac{U}{2} \sum_x \left[\sum_{\sigma=1}^{2} \left(c^\dagger_{x\sigma} c_{x\sigma} - 1/2 \right) \right]^2 . \tag{13}$$

In order to bring the general Hamiltonian (1) to this form, we set

$$\begin{aligned} N_{fl} &= 1 \\ N_{col} &= 2 \\ T_{xy} &= -t\delta_{\langle x,y \rangle} \\ M &= N_{dim} \\ U_k &= -\frac{U}{2} \\ V^{(ks)}_{xy} &= \delta_{x,y}\delta_{x,k} \\ \alpha_{ks} &= \frac{1}{2} . \end{aligned} \tag{14}$$

Note that in this example $N_{dim} = N_{unit\ cell} = Latt\%N$. And since $N_{fl} = 1$ for $SU(N)$-symmetric Hubbard models, we will drop the flavor index σ in the following.

4.1 Hopping term

The hopping matrix is implemented as follows. We allocate an array of dimension 1×1, called `Op_T`. It therefore contains only a single `Operator` structure. We set the effective dimension for the hopping term: $N_{eff} = N_{dim}$. And we allocate and initialize this structure by a single call to the subroutine `Op_make`:

```
call Op_make(Op_T(1,1),Ndim)
```

Since the effective dimension is identical to the total dimension, it follows trivially, that $\boldsymbol{P}_T = \mathbb{1}$ and $\boldsymbol{O}_T = \boldsymbol{T}$. Note that although a checkerboard decomposition is not yet used for the Hubbard model, in principle it can be implemented.

4.2 Interaction term

To implement this interaction, we allocate an array of `Operator` structures. The array is called `Op_V` and has dimensions $N_{dim} \times N_{fl} = N_{dim} \times 1$. We set the effective dimension for the interaction term: $N_{eff} = 1$. And we allocate and initialize this array of structures by repeatedly calling the subroutine `Op_make`:

```
N_dim = Latt%N
N_fl = 1
N_eff = 1

do nf = 1, N_FL
do i  = 1, Latt%N
call Op_make(Op_V(i,nf),N_eff)
enddo
enddo
```

For each lattice site i, the projection matrices $\boldsymbol{P}_V^{(i)}$ are of dimension $1 \times N_{dim}$ and have one non-vanishing entry: $(P_V^{(i)})_{1j} = \delta_{ij}$. The effective matrices are scalars in this example: $\boldsymbol{O}_V^{(i)} = 1$.

Name of variable in the code	Description
`Ndim`	Spacial dimension of the lattice (total number of sites)
`Latt%N`	Number of unit cells of the underlying Bravais lattice
`Op_T`	Array of structure variables that bundles all variables needed to define the hopping operator.
`Op_V`	Array of structure variables that bundles all variables needed to define the two-particle interaction operator.
`N_sun`	Number of fermion colors spin states of the $SU(N_{sun})$-symmetric fermions
`N_fl`	Number of fermion flavors

Table 2: Common variables that are set in the Hamiltonian, operator and lattice modules of the code. !!! We have a missmatch in the labelling: N_{col} = `N_sun` !!!

4.3 Definition of the square lattice

This is set in the subroutine `Ham_latt`. The square lattice is already implemented. In principle, one can specify other lattice geometries and use them by specifying the keyword `Lattice_type` in the parameter file.

4.4 Observables for the Hubbard model

To do next:

- dicuss the measurements: what observables exit and how do I add a new one?
- discuss the implementation of the lattice.
- discuss the Hubbard-Stratonovich decompositions (this is related to the coupling in the operator structure), discuss also the spin-symmetry-breaking HS-decomposition for the Hubbard model.

5 Tutorial: set up a model Hamiltonian

based on the (not yet existing) template `Hamiltonian_template.f90`.