

The *ALF* (Algorithms for Lattice Fermions) project release 1

Tutorial

Florian Goth, Johannes S. Hofmann, Jonas Schwab, Fakher F. Assaad

October 21, 2018

Downloading the code and tutorial

To download the code, type `git clone git@git.physik.uni-wuerzburg.de:ALF/ALF_code.git` in a shell.

To download the tutorial including solutions type:

`git clone git@git.physik.uni-wuerzburg.de:ALF/ALF_Tutorial.git` again in a shell.

Exercise 1) Testing against ED

Run the code with the Mz choice of Hubbard Stratonovitch transformation on a four site ring, at $U/t = 4$ and inverse temperature $\beta t = 2$. For this set of parameters, the exact internal energy reads:

$$\langle -t \sum_{\langle i,j \rangle, \sigma} c_{i,\sigma}^\dagger c_{j,\sigma} + U \sum_{i=1}^N n_{i,\uparrow} n_{i,\downarrow} \rangle = -1.47261997t \quad (1)$$

To reproduce this result you will have to carry out a systematic $\Delta\tau t$ extrapolation keeping $\Delta\tau t L_{\text{Trotter}} = 2$ constant. Recall that the formulation of the auxiliary field QMC approach is based on the Trotter decomposition

$$e^{-\Delta\tau(A+B)} = e^{-\Delta\tau A} e^{-\Delta\tau B} + \mathcal{O}(\Delta\tau^2) \quad (2)$$

The overall error produced by this approximation is of the order $\Delta\tau^2$.

All the files you will need to run the code can be found in the directory `Exercise_1/Start`. Beware to define one directory per run so as to avoid conflicts between different parameter sets. Such conflicts will lead to crashes. Here is the parameter file you should use

```
!=====
! Variables for the Hubb program
!-----
&VAR_lattice
L1 = 4           ! Length in direction a_1
L2 = 1           ! Length in direction a_2
Lattice_type = "Square" ! a_1 = (1,0), a_2=(0,1), Norb=1, N_coord=2
!Lattice_type = "Honeycomb"! a_1 = (1,0), a_2 = (1/2,sqrt(3)/2), Norb=2, N_coord=3
!Model = "Hubbard_SU2" ! Sets Nf=1, N_sun=2. HS field couples to the density
Model = "Hubbard_Mz" ! Sets Nf=2, N_sun=1. HS field couples to the
! z-component of magnetization.
!Model="Hubbard_SU2_Ising"! Sets Nf_1, N_sun=2 and runs only for the square lattice
! Hubbard model coupled to transverse Ising field
/
&VAR_Hubbard ! Variables for the Hubbard model
ham_T = 1.DO ! Hopping parameter
ham_chem= 0.DO ! chemical potential
```

```

ham_U = 4.D0      ! Hubbard interaction
Beta  = 2.D0      ! inverse temperature
dtau  = 0.1D0     ! Thereby Ltrot=Beta/dtau
/

&VAR_QMC          ! Variables for the QMC run
Nwrap = 10        ! Stabilization. Green functions will be computed from
                  ! scratch after each time interval Nwrap*Dtau
NSweep = 1000    ! Number of sweeps
NBin   = 10       ! Number of bins
Ltau   = 1        ! 1 for calculation of time displaced Green functions;
                  ! 0 otherwise
LOBS_ST = 1      ! Start measurements at time slice LOBS_ST
LOBS_EN = 100    ! End measurements at time slice LOBS_EN
CPU_MAX = 0.1     ! Code will stop after CPU_MAX hours.
                  ! If not specified, code will stop after Nbin bins.
/

&VAR_errors       ! Variables for analysis programs
n_skip = 2        ! Number of bins that will be skipped.
N_rebin = 1       ! Rebinning
N_Cov = 0         ! If set to 1 covariance will be computed
                  ! for unequal time correlation functions.
/

```

After running the program, `$DIR/Prog/Examples.out`, run the analysis programs, `bash analysis.sh`. Note that you have to set the environment variable `$DIR` to the directory where the package is located. This can be done by running `./sentenv.sh` in the home directory of the ALF. You will find the internal energy in the file `Ener_scalJ`. This file contains two entries: `OBS: 1` corresponds to the energy and `OBS: 2` to the average sign. If you want to improve the quality of the data, you can rerun the code. Before doing this be sure to move the `confout_*` files to the `confin_*` files. The bash script `out_to_in.sh` will do this for you. Here is the result you should obtain when choosing $\Delta\tau t = 0.05, 0.1, 0.125$.

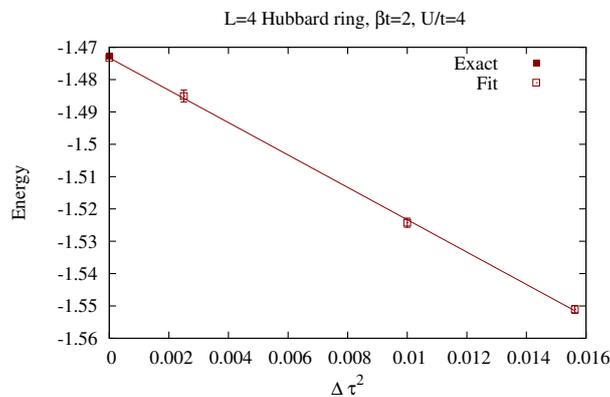


Figure 1: $\Delta\tau t$ extrapolation for the 4-site Hubbard ring.

Exercise 2) Dimensional crossover.

a) Modifying the hopping

Here we will modify the code so as to allow for different hopping matrix elements along the x and y directions of a square lattice. To do so, one merely has to do the following

- Add an extra variable, `Ham_Ty`, in the parameter file in the `VAR_Hubbard` name space
- Declare the variable `Ham_Ty` in the `Hamiltonian_Examples.f90` .
- Read in this variable in `Ham_set` subroutine of the `Hamiltonian_Examples.f90` file.
- Modify the hopping matrix in the subroutine `Ham_Hop` in the `Hamiltonian_Examples.f90` file.

```
DO I = 1, Latt%N
  I1 = Latt%nnlist(I,1,0)
  I2 = Latt%nnlist(I,0,1)
  Op_T(nc,n)%0(I,I1) = cmplx(-Ham_T, 0.d0, kind(0.DO))
  Op_T(nc,n)%0(I1,I) = cmplx(-Ham_T, 0.d0, kind(0.DO))
  !!!!! Modifications for Exercise 2 (a)
  !Op_T(nc,n)%0(I,I2) = cmplx(-Ham_T, 0.d0, kind(0.DO))
  !Op_T(nc,n)%0(I2,I) = cmplx(-Ham_T, 0.d0, kind(0.DO))
  Op_T(nc,n)%0(I,I2) = cmplx(-Ham_Ty, 0.d0, kind(0.DO))
  Op_T(nc,n)%0(I2,I) = cmplx(-Ham_Ty, 0.d0, kind(0.DO))
  !!!!!
  Op_T(nc,n)%0(I ,I) = cmplx(-Ham_chem, 0.d0, kind(0.DO))
ENDDO
```

In the directory `Solutions/Exercise_2` we have duplicated the ALF and commented the changes that have to be carried out to the file `Hamiltonian_Examples.f90` in the `Prog` directory.

As an application of this code, one can consider a ladder system, defined by the parameter file:

```
!=====
! Variables for the Hubb program
!-----

&VAR_lattice
L1 = 14
L2 = 2
Lattice_type = "Square"
Model = "Hubbard_Mz" ! Hubbard_SU2, Hubbard_Mz
/

&VAR_Hubbard
ham_T = 1.0D0
!!!!!! Modifications for Exercise 2
ham_Ty = 2.0D0
!!!!!!
ham_chem= 0.0D0
ham_U = 4.0
Beta = 10.0
dtau = 0.1
/

&VAR_QMC
Nwrap = 10
NSweep = 100
NBin = 10
Ltau = 1
LOBS_ST = 1
```

```

LOBS_EN = 100
CPU_MAX = 0.1
/

&VAR_errors
n_skip = 2
N_rebin = 1
N_Cov = 0
/
! slash terminates namelist statement - DO NOT REMOVE

```

When running the code for the above ladder system and analyzing the spin correlation functions, `SpinZ_eqJR` and `SpinXY_eqJR` one will notice that it is hard to restore the SU(2) spin symmetry and that very long runs are required to obtain the desired equality

$$\langle S_i^z S_j^z \rangle = \langle S_i^y S_j^y \rangle = \langle S_i^x S_j^x \rangle. \quad (3)$$

The structure of the output files `SpinZ_eqJR` and `SpinXY_eqJR` is described in the documentation. For the Mz Hubbard-Stratonovitch transformation it is hence better to consider the improved estimator

$$\langle \vec{S}_i \cdot \vec{S}_j \rangle \quad (4)$$

to compute the spin-spin correlations.

(b) Adding a new observable

Here the aim is to include the new observable equal time observable $\langle \vec{S}_i \cdot \vec{S}_j \rangle$ in the `Hubbard_Mz` code. To achieve this, you will have to carry out the following steps.

- In the subroutine `Alloc_obs` in the `Hamiltonian_example.f90` file you will have to add a new equal time observable with a call to `Call_Obser_Latt_make(Obs_eq(I),Ns,Nt,No,Filename)` with `Ns = Latt%N`; `No = Norb`; `Filename = "SpinT"`, `Nt=1`, `I=5`
- In the subroutine `Obser` you will have to add the Wick decomposition of this observable.

In the program `Hamiltonian_Examples.f90` to be found in the directory `Solutions/Exercise_2/Prog/` we have commented the changes that have to be carried out to add this observable. The new variable takes the name `SpinT` and the results you should obtain are summarized in Fig. 2.

(c) The SU(2) Hubbard Stratonovitch transformation

The SU(2) Hubbard Stratonovitch decomposition, conserves spin rotational symmetry. Run the ladder code with the SU(2) flag in the parameter file switched on (i.e. `Model = Hubbard.SU2`) and compare results.

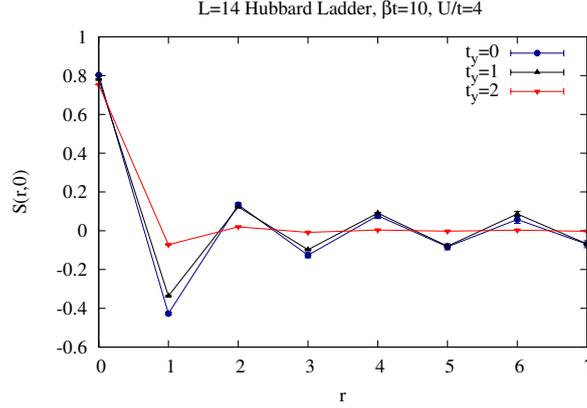


Figure 2: Spin correlation functions along one leg for the Hubbard ladder. As t_y grows the spin gap becomes large enough so as to detect the exponential decay of the spin correlation function on this small lattice size. The underlying physics of odd-even ladder systems is introduced in the article: Elbio Dagotto and T. M. Rice, Surprises on the way from one- to two-dimensional quantum magnets: The ladder materials, *Science* 271 (1996), no. 5249, 618-623.

Exercise 3) Defining a new model: The one-dimensional t-V model.

(a)

In this section, one we will show what modifications have to be carried out for computing the physics of the one dimensional t-V model of spinless fermions.

$$H = -t \sum_i \left(c_i^\dagger c_{i+a} + c_{i+a}^\dagger c_i \right) - \frac{V}{2} \sum_i \left(c_i^\dagger c_{i+a} + c_{i+a}^\dagger c_i \right)^2 \quad (5)$$

The above form is readily included in the ALF since the interaction is written in terms of a perfect square. Expanding the square yields (up to a constant) the desired model:

$$H = -t \sum_i \left(c_i^\dagger c_{i+a} + c_{i+a}^\dagger c_i \right) + V \sum_i (n_i - 1/2) (n_{i+a} - 1/2) \quad (6)$$

In the directory `Solutions/Exercise_3` we have duplicated the ALF and commented the changes that have to be carried out to the file `Hamiltonian.Examples.f90` in the `Prog` directory so as to include the t-V model. Here are the steps to be carried out.

- Add the $t - V$ name space in the parameter file so as to read in the appropriate variables.
- Declare new variables in the `Hamiltonian.Examples.f90` file.
- In the `Ham_set` subroutine of the file `Hamiltonian.Examples.f90` set and read in the parameters for the new model, `t_V`. For this model `NF=1` and `N_SUN=1` since we are working with spinless fermions
- In the `Ham_V` subroutine you will have to add the new interaction. For a given bond at a given time-slice, we need to decouple the interaction:

$$e^{\Delta\tau \frac{V}{2} (c_i^\dagger c_{i+a} + c_{i+a}^\dagger c_i)^2} = \sum_{l=\pm 1, \pm 2} \gamma_l e^{\sqrt{\Delta\tau \frac{V}{2}} \eta_l (c_i^\dagger c_{i+a} + c_{i+a}^\dagger c_i)} = \sum_{l=\pm 1, \pm 2} \gamma_l e^{g \eta_l (c_i^\dagger, c_{i+a}^\dagger) O(c_i, c_{i+a})^T} \quad (7)$$

Here is how this translates in the code.

```
Allocate(Op_V(Latt%N,N_FL))
do nf = 1,N_FL
  do i = 1, N_coord*Ndim
    call Op_make(Op_V(i,nf),2)
```

```

    enddo
  enddo
  Do nc = 1, Latt%N ! Runs over bonds = # of lattice sites in one-dimension.
    I1 = nc
    I2 = Latt%nnlist(I1,1,0)
    Op_V(nc,1)%P(1) = I1
    Op_V(nc,1)%P(2) = I2
    Op_V(nc,1)%O(1,2) = cmplx(1.d0 ,0.d0, kind(0.D0))
    Op_V(nc,1)%O(2,1) = cmplx(1.d0 ,0.d0, kind(0.D0))
    Op_V(nc,1)%g      = SQRT(CMPLX( DTAU*Ham_Vint/2.d0, 0.D0, kind(0.D0)))
    Op_V(nc,1)%alpha  = cmplx(0d0,0.d0, kind(0.D0))
    Op_V(nc,1)%type =2
    Call Op_set( Op_V(nc,1) )
  enddo

```

- Finally you will have to update the `Obser` and `ObserT` routines for the calculation of the equal and time displaced correlations. For the `t_V` model you can essentially use the same observables as for the `Hubbard_SU(2)` model.

You can now run the code for various values of V/t . A Jordan-Wigner transformation will map the `t_V` model onto the XXZ chain:

$$H = J_{xx} \sum_i S_i^x S_{i+a}^x + S_i^y S_{i+a}^y + J_{zz} \sum_i S_i^z S_{i+a}^z \quad (8)$$

with $J_{zz} = V$ and $J_{xx} = 2t$. Hence when $V/t = 2$ we reproduce the Heisenberg model. For $V/t > 2$ the model is in the Ising regime with long-range charge density wave order and is an insulator. In the regime $-2 < V/t < 2$ the model is metallic and corresponds to a Luttinger liquid. Finally, at $V/t < -2$ phase separation between hole rich and electron rich phases occur. Fig. 3 shows typical results.

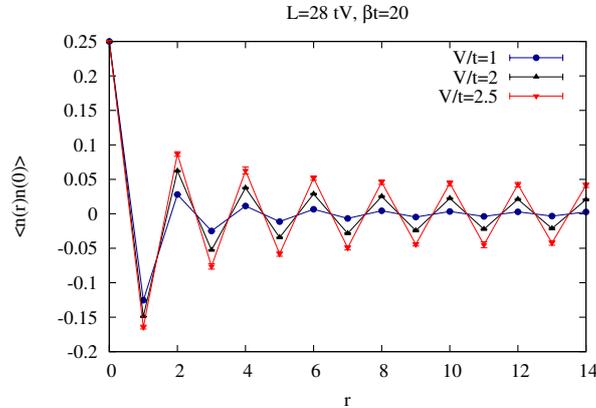


Figure 3: Density-Density correlation functions of the `t_V` model. In the Luttinger liquid phase, $-2 < V/t < 2$ it is known that the density -density correlations decay as $\langle n(r)n(0) \rangle \propto \cos(\pi r) r^{-(1+K_\rho)}$ with $(1 + K_\rho)^{-1} = \frac{1}{2} + \frac{1}{\pi} \arcsin\left(\frac{V}{2|t|}\right)$ (A. Luther and I. Peschel, Calculation of critical exponents in two dimensions from quantum field theory in one dimension, Phys. Rev. B 12 (1975), 3908.) The interested reader can try to reproduce this result.

(b)

How would you use the code to carry out simulations at $V/t < 0$?