Mapping Out Bose-Hubbard Model's Phase Diagram

Yasamin Khorramzadeh
Department of Physics, Virginia Tech, Blacksburg, Virginia 24061

Experiments with cold atoms trapped in optical lattices offer the potential to realize a variety of novel phases but suffer from severe spatial inhomogeneity that can obscure signatures of new phases of matter and phase boundaries. Experimentally, the challenge of obtaining the phase diagram at .nite T lies in identifying measurable properties that can diagnose di.erent phases. We propose an observable named Core Compressibility κ_c that can be measured in experiments and explore the phase diagram of the Bose-Hubbard model.

First we focus on Total Compressibility κ which can distinguish different phases but it is not measurable in experiment. Then we use a Quantum Monte Carlo(QMC) Simulation to make a comparison between Core Compressibilityκ_c and Total Compressibilityκ and we find that κ_c is essentially identical to κ when system is in certain states, and therefore the measure of the κ_c offers a valuable tool for mapping out phase diagrams which can be observed in experiments as well.(Fig 1)

The QMC methods represent a powerful and broadly applicable computational tool for finding very accurate solutions of the systems with many coupled degrees of freedom.The algorithms are intrinsically parallel and are able to take full advantage of the present-day high-performance computing systems. To conduct QMC simulation we used ALPS (Algorithms and Libraries for Physics Simulations) which is an open source effort aiming at providing high-end simulation codes for strongly correlated quantum mechanical systems. Implementing ALPS we run the simulation on parallel machines using MPI on the computer cluster provided at Virginia Tech department of Physics.

In practice several quantities are accessible using different ALPS Applications including observables : Energy, Specific heat, Susceptibilities, Struc-

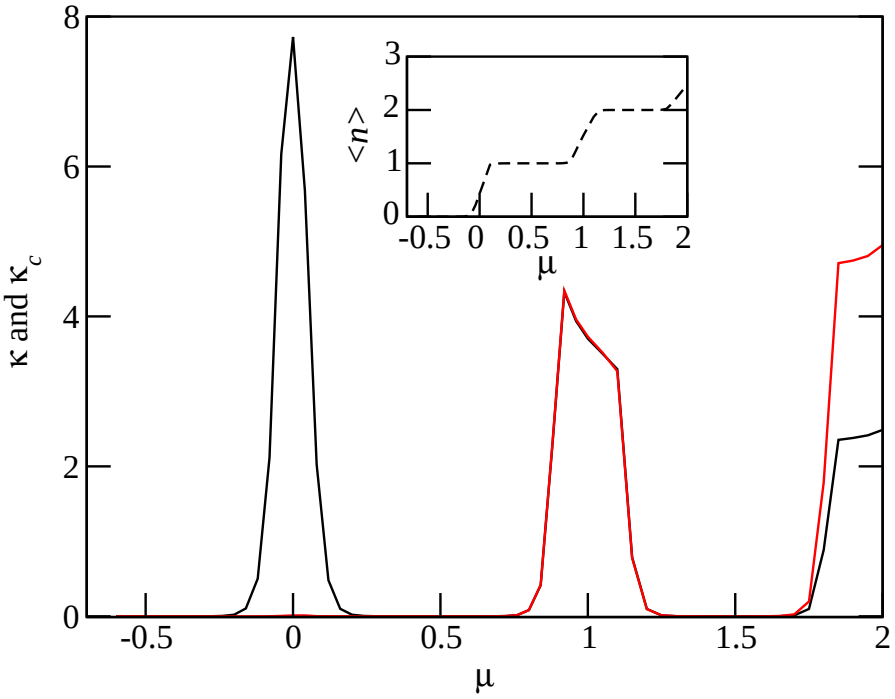

Figure 1: Compressibility κ(Black line) and core compressibility κ_c(Red line) versus chemical potential in a uniform system. The two lines merge when there is a little triple occupancy. The inset plot shows density as a function of μ.

ture factor, Spin stiffness, Green functions ,etc. In some cases, one can define its own desired observable. Using this ability of the ALPS we defined new measurements that are required for calculating Core and Total Compressibilities. In practice we start with defining new measurements then we run one of ALPS.s built in applications to create primary data which includes both program and user defined measurements.(Q_1 in figure 2)Then we check whether it is done successfully or not. (Q_2) If the job has been stopped before it finished for reasons such as exceeding the expected wall time or an unknown reason we need to re run the job, however we do not need to start from scratch as ALPS would store the current status of the job in a separate

file. To restart the simulation we have to change the submission command so that it is using this file. (Q_4) When we got the required data (Q_3) we can continue by running another application to calculate Core and Total K based on data of the previous jobs.(Q_8) Again checking whether our job is finished successfully or not (Q_9) and resubmit it (Q_{11}) till we get the final data files. The simulation output files only contain the collected measurements from all runs. Details about the individual Monte Carlo runs for each simulation can be extracted using ALPS tools. Therefore in the last step we collect our desired data using ALPS.s different tools.(Q_{11})

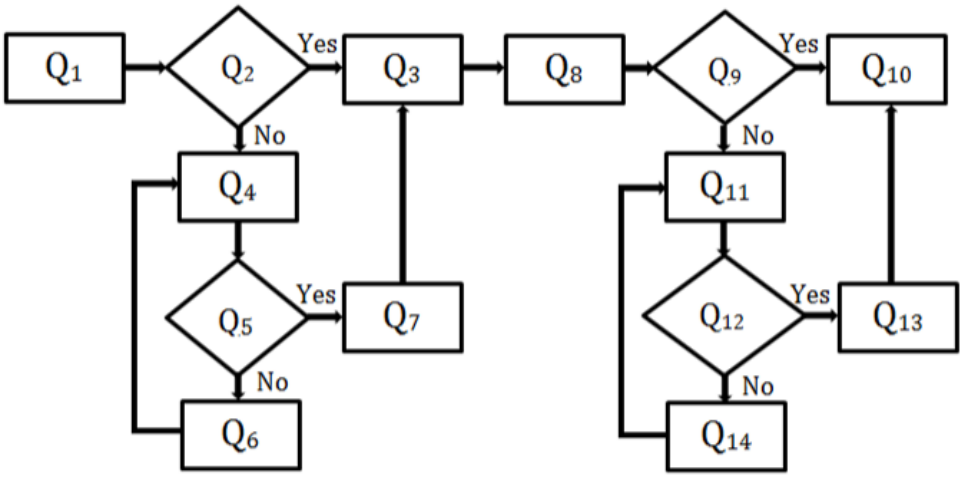

Figure 2: Flow Chart shows the hierarchy of the process to find κ and κ_c for a single set of parameters.

This process needs to be repeated for different sets of parameters. To summarize I can say that for an individual set of parameters I have multiple jobs that need to be run in a consecutive order in Figure 2 shows the flowchart of this process. Then we need to repeat the process for several different sets of parameters.

One practical concept that I have learned in the OSG summer school was DagMan which automatically controls the work flow of jobs when we have several hierarchical tasks. Although Condor's DagMan is not available on the computer cluster that I am using but one can apply this idea to his own problem and write a code that can manage the order of tasks that needs to be

done to get the desired data. For example the following script is a simplified routin that implements the DagMan manager for our problem. Using this method we benefit from the check pointing ability of ALPS program by going through stepsQ_4 and Q_{11} in Fig2.

```
#!/bin/sh

for ((  i = 1 ;  i <= 10;  i++  ))
do
 parameter2xml parmset_$i
 qsub submpijob_$i
done

sleep $wall_time

for ((  i = 1 ;  i <= 10;  i++  ))
do
 MY_MESSAGE=‘find . -name "submpijob_$i.e*" -exec grep
 "Finished with all tasks." ’{}’ \;‘
 THIS_MESSAGE="Finished with all tasks."

 if [ "$MY_MESSAGE" == "$THIS_MESSAGE" ]; then
     extracttext DesiredOutPut.xml parmset_$i.out
     parameter2xml parmset_$i.out
     qsub submpijob_modified
 else
     sed -e   ’s/in.xml/out.xml/g’ submpijob_$i > re_submpijob_$i
     qsub re_submpijob_$i
 fi
done
```

Running several simulations simultanously provides a great tool to study Condensed matter physics which is dealing with phenomena that arise when large number of particlesn are brought together. I would like to thank the OSG summer school organizers and Instructors for giving me the opportunity to learn more about High Throughout Programming technique.

References

[1] M. Greiner, Ph.D. thesis, Ludwig-Maximilians-Universitt Munchen, 2003