Optimizing Pre-Formed Molecules in Mixtures of Ultracold ^{40}K and ^{87}Rb on an Optical Lattice: A Challenge Grant and Capabilities Application Project

J.K. Freericks
Department of Physics, Georgetown University, Washington, DC
freericks@physics.georgetown.edu

Abstract

This work is part of a Defense Advanced Research Projects Agency (DARPA) sponsored project to build an optical lattice emulator, where models of strongly correlated electrons in condensed matter physics are simulated with ultracold atoms moving in an optical lattice. Recently, members of our team have been able to form dense clouds of dipolar fermionic molecules from mixtures of (fermionic) ^{40}K and (bosonic) ^{87}Rb. Here, we use high performance computing (HPC) resources to find a way to improve the efficiency of molecule formation from the current 20% to almost 100% and thereby show how to create a much denser cloud of dipolar molecules. Our code scales nearly linearly on up to 4,000 (or more) processors, and runs at an efficiency that is almost at 100% of the speed for one arithmetic operation per clock cycle. We have not been able to get the code to run with multiple operations per clock cycle in spite of using highly efficient and optimized libraries for BLAS and LAPACK.

1. Introduction

Recent experimental work on mixtures of ultracold ^{40}K (fermionic) and ^{87}Rb (bosonic) has shown how to bind the atoms into weakly bound molecules via a sweep of the magnetic field and then employ a complicated pulse sequence of light (called stimulated Raman adiabatic passage or STIRAP) to make a transition from the weakly bound molecular state to the tightly bound ground state, which is dipolar[1]. Being able to create a quantum degenerate gas of molecular dipoles has been a long quest for experimental researchers. This novel state of matter has the potential for illustrating new and unique quantum-mechanical effects. It is believed that these systems might be able to illustrate topological quantum phases due to the long-range interaction of the dipoles. In addition, if the system becomes quantum coherent, it may have unique properties in response to electric fields which could ultimately be used as ultrasensitive electric field sensors (similar to how superconductivity led to the SQUID as an ultrasensitive sensor for magnetic fields). Finally, these systems could also be employed as quantum computers.

Unlike ordinary computers, quantum computers have the potential to revolutionize the way that certain types of problems are solved. For example, a quantum computer can factorize large numbers into their prime factors much more efficiently than a conventional computer can, which would require a change in the way that standard encryption technology is employed. It also could allow for completely secure transmission of one-time key pads for encrypted communication purposes. But Defense Advanced Research Projects Agency's (DARPA) interest in this problem lies in another direction. They want to use quantum computers as analog simulators of complex quantum-mechanical systems (ultimately with the goal of speeding up the process for novel materials design). This idea was first described in 1982 by Richard Feynman[2]. He proposed to create artificial, but well controlled, quantum-mechanical systems that could mimic the behavior of real physical systems. By performing experiments on these artificial systems, one could determine how the quantum systems behave and learn how to tune or optimize properties of interest.

One of the major technological breakthroughs of the past few decades has been the technology of laser cooling (which was awarded a Nobel Prize in 1997). By shining light that is tuned close to an atomic resonance, one can create a situation where atoms moving toward the light can absorb it and be pushed back, but those moving away cannot absorb the light (via the Doppler effect) and thereby reduce the average velocity of the atoms. Next, the atoms are further cooled by letting the most energetic atoms evaporate out of the trap, and leaving behind only the coldest atoms. The field of ultracold atomic physics has created a wealth of new experimental work on these systems. The final important advance for the work we discuss here is the creation of an optical lattice by making

978-0-7695-3946-1/10 $26.00 © 2010 IEEE
DOI 10.1109/HPCMP-UGC.2009.35

a standing wave with a laser which results in a corrugated potential that the atoms move in. If their energy is low enough and the corrugation deep enough, then the atoms act like quantum particles that tunnel between neighboring sites and interact with each other when on the same lattice site.

This physical system, constructed from a set of atoms which move between nearest neighbors on a lattice and interact when two particles are located on the same lattice site, is reminiscent of the simplified models used in solid state physics to describe strongly correlated electrons in a crystal (like the Hubbard model[3] or the Falicov-Kimball model[4]). The atoms act like electrons (if they are fermions), but now move on a much larger sized lattice (microns rather than nanometers), allowing for easier access to study their properties. Furthermore, because the particles have significant internal structure, there is a range of new probes that can be used to examine their behavior.

We focus on Bose-Fermi mixtures of rubidium with potassium. The ^{87}Rb is more than twice as heavy as the ^{40}K, and we will work in a regime where the motion of the Rb relative to the K is sharply suppressed due to the larger mass. In particular, we ensure that the hopping of the K is at least ten times larger than the hopping of the Rb, so that we can neglect the quantum-mechanical effects of the Rb motion. But, we assume that the system can explore all possible positions for the heavy atoms (at least during the period where the optical lattice is initially turned on, but possibly also during the period of the experiment when the optical lattice is at its full depth), and hence we analyze the problem by using an annealed statistical ensemble, like in the Ising model for magnetism. This system is described by the Falicov-Kimball (FK) model[4] which has two kinds of particles: itinerant spinless fermionic K atoms with creation and annihilation operators $c_i^\dagger$ and c_i for the K atoms at site i (located at position $\mathbf{R}_i$) and localized spinless bosonic Rb atoms with the corresponding operators $b_i^\dagger$ and b_i. The fermionic operators satisfy canonical anticommutation relations $\left\{c_i^\dagger, c_j\right\}_+ = \delta_{ij}$ while the bosonic operators satisfy canonical commutation relations $\left\{b_i^\dagger, b_j\right\}_+ = \delta_{ij}$ The Bose-Fermi FK Hamiltonian is

$$\begin{aligned}\mathcal{H} = &-\sum_{ij} t_{ij} c_i^\dagger c_j + U_{bf}\sum_i c_i^\dagger c_i b_i^\dagger b_i \\ &+\frac{1}{2}U_{bb}\sum_i b_i^\dagger b_i\left(b_i^\dagger b_i - 1\right) \\ &+\sum_i (V_i - \mu) c_i^\dagger c_i + \sum_i \left(V_i^b + E_b\right) b_i^\dagger b_i,\end{aligned} \tag{1}$$

where $-t_{ij}=-t$ is the nearest-neighbor hopping matrix, $U_{bf} < 0$ is the on-site attraction between K and Rb atoms (we choose an attraction because we want to optimize the probability to find a K sitting in the same lattice site as a Rb and because this interaction can be tuned by an interspecies Feshbach resonance) $U_{bb}>0$ is the Rb-Rb repulsion, μ is the chemical potential of the K atoms and E_b is the local site energy of the Rb atoms. The two potentials V_i and V_i^b represent the additional trap, which is chosen to be harmonic for both species but (in principle) with different tunable curvatures. We parametrize the traps with an effective length R and R^b where $V_i - tR_i^2/R^2$ and $V_i^b - tR_i^2/R^{b2}$. Because the polarizabilities of K and Rb are so close, we choose the trap curvatures to be equal (assuming an optical trap) and to be characterized by the length corresponding to eleven lattice spacings ($R = R^b = 11$).

We will simulate a system on a 51×51 square lattice with 625±10 light atoms and 625±10 heavy atoms. The two chemical potentials μ and $-E_b$ must be adjusted to produce the correct filling at each temperature for which the simulation is run. We choose two lattice depths for the optical lattice potential (these depths must be deep enough that the single-band model applies to the system). The first depth is 15 recoil energies (expressed in terms of the ^{87}Rb recoil energy E_R) and the second depth is 20 recoil energies (these are the depths in the x–y plane; we choose the depth along the z-axis to be $40E_R$ so we have well separated planes). For the $15E_R$ depth lattice, we have the K hopping is 190 Hz, while the Rb hopping is 14 Hz (and will be neglected). The Bose-Bose repulsion satisfies U_{bb}/t=5.70. The Bose-Fermi attraction can be tuned via a magnetic interspecies Feshbach resonance, and is chosen to satisfy U_{bf}/t=−2, −6, and −10 here. For the $20E_R$ case, we have the K hopping is 109 Hz and the Rb hopping is 5.4 Hz (and is neglected). The Rb-Rb repulsion is U_{bb}/t=11.52. We pick the Bose-Fermi attraction to be U_{bf}/t=−8, −12, and −16.

There are two main quantities we are interested in calculating. The first is the efficiency for pre-formed molecules, defined to be the probability to find a K atom on the same lattice site as a Rb atom, and defined to be the expectation value

$$E = \frac{\left\langle \sum_i c_i^\dagger c_i b_i^\dagger b_i \mathcal{P} \right\rangle}{\min\left[\sum_i \left\langle c_i^\dagger c_i \right\rangle, \sum_i \left\langle b_i^\dagger b_i \right\rangle\right]}, \tag{2}$$

where the angular brackets denote the conventional trace over all states weighted by the statistical density matrix $\exp[-\beta\mathcal{H}]/\mathcal{Z}$ (with $\mathcal{Z}$=Tr $\exp[-\beta\mathcal{H}]$ the partition function and $\beta=1/T$ the inverse temperature), and the operator $\mathcal{P}$ projects onto the subspace with no sites that have two or more bosons per site. This expectation value can be calculated directly from the Green's functions on the imaginary axis, as described below. The second is the

entropy per particle, which can be found most easily via an integral over the many-body density of states, and requires a real-axis calculation of the Green's functions. It is the latter calculation that requires the most significant CPU time to complete.

2. Algorithm

The computational approach is based on inhomogeneous dynamical mean-field theory (IDMFT), which is a generalization of the successful DMFT[6] to inhomogeneous systems. This generalization was first worked out for multilayered systems by Potthoff and Nolting[7] and is now summarized in a book[8]. More recently, Tran applied it to ultracold atoms in traps[9], and since then much work has followed (see, for example, References 10 and 11).

The IDMFT approach begins with the quantum-mechanical Green's functions for the fermions, which are defined, for imaginary time τ, as

$$G_{ij}(\tau) = -\mathrm{Tr} e^{-\beta\mathcal{H}} \mathcal{T}_\tau c_i(\tau) c_j^\dagger(0) \frac{1}{\mathcal{Z}} = -\left\langle c_i(\tau) c_j^\dagger(0) \right\rangle, \quad (3)$$

where Tr denotes the trace over all many-body eigenstates and $\mathcal{T}_\tau$ is the time-ordering operator, which moves earlier times to the right (with a sign change if two fermionic operators are interchanged). The time-dependent fermionic creation and annihilation operators are written in the Heisenberg picture as

$$c_i(\tau) - e^{\tau\mathcal{H}} c_i e^{-\tau\mathcal{H}} \text{ and } \mathrm{c}_i^\dagger(\tau) - e^{\tau\mathcal{H}} \mathrm{c}_i^\dagger e^{-\tau\mathcal{H}}, \quad (4)$$

for the imaginary time τ. By using the invariance of the trace under the interchange of the product of two operators, one can show that the Green's function is antiperiodic in τ over the range $0 \le \tau \le \beta$, so one can describe the Green's function by a Fourier series using the Matsubara frequencies $i\omega_n = i\pi T\,(2n+1)$ for n an integer (which corresponds to extending the range of the imaginary time to all values, but preserving the antiperiodicity). So we have

$$G_{ij}(i\omega_n) = \int_0^\beta d\tau e^{i\omega_n \tau} G_{ij}(\tau). \quad (5)$$

Using an equation of motion, found by differentiating the Green's function with respect to τ, then yields the equation

$$\sum_k \left[\left\{ i\omega_n + \mu - V_i - \Sigma_i(i\omega_n) \right\} \delta_{ik} + t_{ik} \right] G_{kj}(i\omega_n) = \delta_{ij}, \quad (6)$$

so the Green's function is found by inverting the matrix defined in the square brackets (because of the inhomogeneity, the matrix can only be inverted numerically). We have introduced the notation $\Sigma_i(i\omega_n)$ for the local self-energy at site i. The self-energy is local (meaning it is diagonal, rather than a matrix in the spatial coordinates) within the DMFT approach, but it can vary from site to site for IDMFT. The self-energy is calculated by solving an effective single-site impurity problem in a time-dependent field that is determined self-consistently. Without going into details, the set of equations that the self-energy satisfies for the FK model is[12,6]

$$G_{ii}^0(i\omega_n) = \frac{1}{G_{ii}^{-1}(i\omega_n) + \Sigma_i(i\omega_n)}, \quad (7)$$

$$G_{ii}(i\omega_n) = \sum_{j=0}^{\infty} \rho_i^b(j) \frac{1}{\left[G_{ii}^0(i\omega_n) \right]^{-1} - jU_{bf}}, \quad (8)$$

and

$$\Sigma_i(i\omega_n) = \left[G_{ii}^0(i\omega_n) \right]^{-1} - G_{ii}^{-1}(i\omega_n). \quad (9)$$

Here, the symbol $G_{ii}^0(i\omega_n)$ is called the effective medium and $\rho_i^b(j)$ is the probability to have j bosonic atoms at site i. The IDMFT algorithm to solve for the Green's function for a given set of parameters is as follows: (i) set the self-energy equal to an initial value on all lattice sites; (ii) calculate the local Green's function from Eqn. 6 at each lattice site i; (iii) determine the effective medium using the local Green's function and the old self-energy in Eqn. 7; (iv) find the new Green's function from Eqn. 8; and (v) find the new self-energy from Eqn. 9 using the new Green's function and the effective medium. Steps (ii)–(v) are repeated until the results stop changing at a fixed point. This can take many thousands of iterations in some cases, but in nearly all cases shown here the number of iterations is less than a few hundred.

In order to carry out the calculation, we still need to determine $\rho_i^b(j)$ at each lattice site. The Rb filling is a functional of the K Green's functions, and is expressed as $\rho_i^b(j) = \mathcal{Z}_i(j) / \Sigma_{j=0}^{\infty} \mathcal{Z}_i(j)$ with

$$\mathcal{Z}_i(j) - 2e^{-\beta\left(E_b + V_i^b\right)j - \beta U_{bb} j(j-1)/2 + \beta\left(\mu - V_i - jU_{bf}\right)/2} \times \prod_{n=-\infty}^{\infty} \frac{\left[G_{ii}^0(i\omega_n) \right]^{-1} - jU_{bf}}{i\omega_n}. \quad (10)$$

Finally, the K atom filling on each site i is found from

$$n_i = T \sum_{n=-\infty}^{\infty} G_{ii}(i\omega_n) + \frac{1}{2}, \quad (11)$$

where special care must be taken to properly regularize the summation.

Since we want to work with a fixed number of K and Rb atoms, we need to run the calculation for a few different values of the chemical potentials μ and $-E_b$, and then adjust them so that one reaches the target particle numbers. Typically, somewhere between two to fifteen

runs are required to get the fillings within the target range of 625±10, although in most cases our fillings are 625±1.

In addition, since the computational size of the problem grows with the number of Matsubara frequencies used in the simulation (as do the memory requirements), we use sum rules for the high frequency behavior of the Green's function, effective medium and self-energy to reduce the number of Matsubara frequencies used in the calculation by about one order of magnitude with no loss in accuracy[13]. We typically use between 64 (T≥0.1) and 1,020 (T=0.05) positive Matsubara frequencies for a given calculation. The general approach for these sum rules is discussed in Reference 13, but here they must be generalized to the case where the heavy particles are bosons. For example, the first moment $\mu_1^R(i,i)$ sum rule for the local Green's function is

$$-\frac{1}{\pi}\int_{-\infty}^{\infty} d\omega\ \omega \mathrm{Im}\, G_{ii}(\omega) = U_{bf}\left\langle b_i^\dagger b_i\right\rangle - \mu + V_i, \qquad (12)$$

and the second moment $\mu_1^R(i,i)$ is

$$\begin{aligned} -\frac{1}{\pi}\int_{-\infty}^{\infty} d\omega\ \omega^2 \mathrm{Im}\, G_{ii}(\omega) &= \sum_j t_{ij}t_{ji} + (\mu - V_i)^2 \\ -2(\mu - V_i)U_{bf}\left\langle b_i^\dagger b_i\right\rangle &+ U_{bf}^2\left\langle b_i^\dagger b_i b_i^\dagger b_i\right\rangle. \end{aligned} \qquad (13)$$

Here $G_{ii}(\omega)$is the analytic continuation of the retarded Green's function to the real axis. We have determined these moment sum rules through third order for the local Green's function and through first order for the self-energy. They are employed to verify the accuracy of the calculations, as described below. In addition, they are used to reduce the number of Matsubara frequencies needed to perform summations over Matsubara frequencies, as in Eqn. 11. To illustrate how this is done, one first defines an approximate Green's function via

$$G_{ii}^{app}(i\omega_n) = \frac{1}{i\omega_n} + \frac{\mu_1^R(i,i)}{(i\omega_n)^2} + \frac{\mu_2^R(i,i)}{(i\omega_n)^3} + \frac{\mu_3^R(i,i)}{(i\omega_n)^4}, \qquad (14)$$

with the exact values of the moments determined, for example, in Eqns. 12 and 13. Then, we find the filling satisfies

$$\begin{aligned} n_i &= \frac{1}{2} - \frac{\beta}{4}\mu_1^R(i,i) + \frac{\beta^3}{48}\mu_3^R(i,i) \\ &+ T\sum_{n=-n_c-1}^{n_c}\left[G_{ii}(i\omega_n) - G_{ii}^{app}(i\omega_n)\right], \end{aligned} \qquad (15)$$

where the summation is over a finite number of Matsubara frequencies determined by the cutoff n_c. Instead of having this sum decay like $1/\omega_n$ as it does in Eqn. 11, here it decays like $1/\omega_n^5$ and hence one can employ a much smaller cutoff. One can use a similar approach to improve the accuracy of the infinite products needed to determine the different contributions to the partition function and to the probabilities for finding j bosons on site i. We don't show those formulas here.

The parallel implementation of the IDMFT algorithm is efficiently done within the master-slave format. The solution of the impurity problem for the self-energy requires only information of the Green's functions at each site. We do need to evaluate the infinite products to find the Rb atom density as in Eqn. 10, but then the remainder is straightforward arithmetic. We have each slave node solve for the self-energy at a given site for all Matsubara frequencies used in the simulation as one step in the parallel implementation. The other step is to send the calculation of the matrix inversion needed to find the local Green's function for each Matsubara frequency to a different slave node. Since in both cases, the communications involve just vectors, rather than matrices, the code rarely encounters communications-based limitations in scaling to large numbers of processors.

The parallel implementation for the IDMFT algorithm is then as follows: 1) the master node initializes all parameters for the calculation and sends them to the slave nodes; 2) the master node loops through the Matsubara frequencies, sending a vector of self-energy values $\Sigma_i(i\omega_n)$ with fixed n to each slave node; 3) the slave nodes perform the matrix inversion and send the local Green's function vector back to the master; 4) once all Matsubara frequency calculations are complete, the master sends each slave node the local Green's function and the self-energy for a fixed lattice site and all Matsubara frequencies (also vectors); and 5) the slave nodes solve the impurity problem to determine the new self-energy and send them back to the master. This procedure is iterated, and when errors are small enough, the calculation stops (our tolerance is usually errors of less than one part in 10^8 for the self-energy at all lattice sites). LAPACK and BLAS routines are used for the matrix operations to maximize the speed and efficiency of the code.

The efficiency E can be determined from the Green's functions on the imaginary axis. It satisfies

$$E = \frac{\Sigma_i \rho_i^b(1) T\Sigma_n\left[G_{ii}^0(i\omega_n)^{-1} - U_{bf}\right]^{-1}}{\min\left[\Sigma_i T\Sigma_n G_{ii}(i\omega_n), \Sigma_i\Sigma_j \rho_i^b(j) j\right]}. \qquad (16)$$

This sum can be evaluated efficiently by using the sum rules to make the summation decay faster than it does without making any adjustments, but the final formula is too long to be included here.

In addition to the Green's functions on the imaginary axis, we also need the Green's functions on the real axis, particularly to determine the local entropy of the system. The local entropy can be found via a simple integration of the local density of states $\rho_i(\omega)$=−Im$G_{ii}(\omega)/\pi$, where

$G_{ii}(\omega)$ is the local Green's function on the real axis. The local entropy density is then

$$s_i = -\sum_{j=0}^{\infty} \rho_i^b(j) \ln \rho_i^b(j) - \int d\omega \rho_i(\omega) \left[f(\omega) \ln f(\omega) + \{1 - f(\omega)\} \ln \{1 - f(\omega)\} \right], \quad (17)$$

with $f(\omega)=1/[1+\exp(\beta\omega)]$ the Fermi-Dirac distribution function.

The real-axis Green's function is found from analogous equations to those used for the imaginary axis, except now we know what the Rb atom densities are at each lattice site, so we do not need to recalculate them during the iterations, and we know the chemical potentials too. Hence, we merely need to set up a grid in frequency space and perform the IDMFT algorithm using a real frequency ω instead of a Matsubara frequency. Since we have a fixed grid of frequencies, the computational size is identical for all temperatures. We typically use 1,250 or 2,500 processors and run for about 200 iterations. All relevant moment sum rules for the Green's functions and self-energies are checked, and they are verified to high accuracy in nearly all cases. We do see errors when the self-energy picks up sharp delta-function-like peaks, because our (coarse) grid will overestimate their contribution to the moments, and we find sum-rule violation for the Green's functions at both high and low temperatures. At high temperatures, significant spectral weight sits beyond the finite range of frequency used in the calculation, while at low temperature, the Green's function develops sharp peaks which are difficult to properly represent on a coarse frequency grid. But neither of these issues is too significant to cause any serious problems.

3. Implementation of Algorithm on HPC Resources

This work has been supported by both a challenge project and a Capabilities Application Project (CAP). The Challenge Project was run primarily on the Engineer Research and Development Center (ERDC) [jade, Cray XT4] and Air Force Research Laboratory, DoD Supercomputing Resource Center (AFRL DSRC) [hawk, Altix] machines, while the CAP was run at Arctic Region Supercomputing Center (ARSC) on pingo (Cray, XT5). The code is highly transportable, and has been run on many different machines, with essentially no modification of the code needed to run on different platforms. The code was written by the user, but does employ both LAPACK and BLAS routines, which encompass the vast majority of the computational time. Since these libraries are always highly optimized for different machines, the portability often maintains the high efficiency and (near) linear scaling.

The IDMFT algorithm uses two main codes. The first, an imaginary axis code, determines the chemical potentials and then the local densities of the K and Rb atoms; it also determines the efficiency for the pre-formed molecules. The second, a real axis code, determines the local density of states, and hence the entropy distribution. Since the two codes are so similar in structure, most of the scaling and performance analysis was performed on the imaginary axis code.

We used a few different techniques to determine the scaling and performance of the codes. The simplest technique we used was just timing of the code. Since the input/output part of the code is infrequent (it is performed every 200 iterations for the imaginary axis code and every 50 iterations for the real axis code) the timings were restricted to the main computational loops of the code, namely the IDMFT algorithm itself. We performed a strong-scaling analysis, where a large problem was run on a series of different numbers of CPUs for a parallel run. By examining how the performance varies as the number of CPUs increases, one can examine how the computational speed is related to the number of CPUs and determine the overall strong scaling performance for the code; for perfect performance, the speed will increase linearly with the number of CPUs. We also examined weak scaling, where one takes the same type of problem, but increases the size of the problem when running on more CPUs and examines the total computational time, which would be a constant for perfect weak scaling. In our code, we easily can increase the code size by merely lowering the temperature and thereby using more Matsubara frequencies in the calculation. These timings appear in Figure 1. The weak scaling results were similarly impressive, but we do not show the plots here.

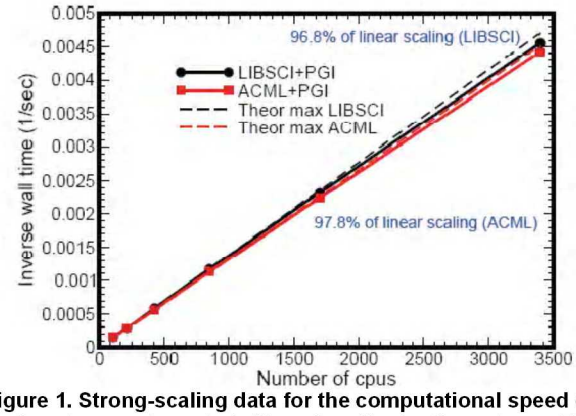

Figure 1. Strong-scaling data for the computational speed of the imaginary axis code (with no input/output) on pingo. The theoretical maximum (dashed line) is found by fitting the computational speed versus number of CPUs for small CPUs and extrapolating the linear curve. Note how the codes are performing well above 95% of linear scaling for over 3,400 CPUs (essentially the full size of the machine). Such performance is outstanding.

We went further than just a scaling analysis though. Linear scaling is not enough to show that a code is utilizing a resource effectively. One can have a code that is inefficient, but nevertheless scales well with the number of CPUs, because the code never pushes the machine to the limits in communications or computation, due to the inefficient way that the code handles data or orders the computational steps. In order to verify the overall *efficiency* of the code, we used the PAPI suite to measure the Gflops of each of the slave nodes to determine how they perform during the main computational loop of the code. We examined both the strong and weak scaling results, but here we focus again on the strong-scaling case in Figure 2. Note that we have essentially no degradation in going to larger numbers of processors, but we are only able to get to about one operation per clock cycle, while the chip is theoretically supposed to be able to perform four operations per clock cycle. The chip runs at 2.3 GHz, and hence can perform 9.2 Gflops if one achieves 4 operations per clock cycle. A performance near 2 Gflops indicates one is achieving almost one operation per clock cycle.

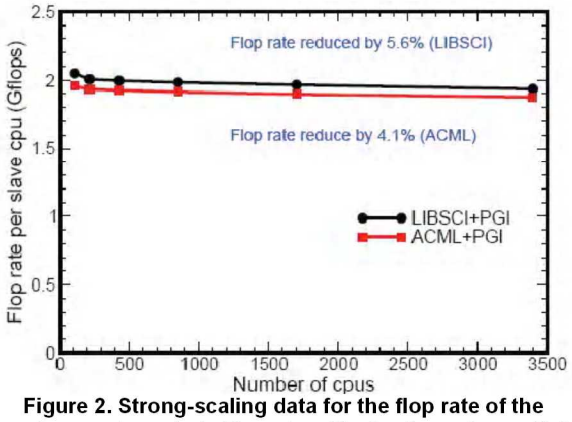

Figure 2. Strong-scaling data for the flop rate of the imaginary axis code (with no input/output) on pingo. Note how the rate remains almost constant up to 3,400 CPUs. The overall rate corresponds to nearly one operation per clock cycle. But this is much less than the theoretical peak flop rate corresponding to four operations per clock cycle. In spite of trying many different compiler options, the performance never improved to higher than one per clock cycle.

We tested two different library options, both LibSci and ACML. The performance was always somewhat better overall for the LibSci library, although there was not a significant difference. The codes are written in FORTRAN, using essentially all FORTRAN-77 calls except for dynamical memory addressing in the real-axis code which requires FORTRAN-90. Our compiler flags (PGI) were set to fast optimization options. We found no significant improvement in performance with any other flag options for the compiler. One can see in Figure 1, that the theoretical maximum speed up (linear curve) for the Lib-Sci case (black line with circles) is better than the ACML case (red line with squares). One can immediately see that the code is giving almost perfect linear scaling up to 3,400 CPUs. The overall performance is much better than 95% of linear scaling, which is outstanding performance for strong scaling. We see similar results for the flop rate in Figure 2. We ran the performance analysis package (PAPI) for the strong scaling cases already shown in Figure 1. We also examined two library packages (LibSci and ACML) for the LAPACk and BLAS subroutines. One can see that the performance is outstanding but we are unable to achieve better than one operation per clock cycle. We also checked the real axis codes, and the performance was similar to the imaginary axis codes.

These codes are highly transportable, and there is similar performance on the other machines used in the challenge project. In total, we have used about 2,500,000 CPU-hrs on the CAP and about 1,500,000 CPU-hrs on the Challenge Project to date.

4. Scientific Results

We briefly discuss some of the scientific results from this work. In Figure 3, we show the efficiency versus temperature for a $20E_R$ depth optical lattice. We set the Bose-Fermi attraction to three different values U_{bf}=−16t, −12t, and −8t. In all cases, the K and Rb rapidly "bind" together to create a pre-formed molecule, that is, the K and Rb atoms prefer to sit on the same lattice site. The efficiency itself does not depend too strongly on the magnitude of the attraction once it is large enough. For example, in the case where the attraction is −16t, and the temperature is equal to the hopping, the efficiency is nearly 100%. Even at a temperature on the order of the bandwidth of 8t, the efficiency is near 45%. One can see that this optical lattice produces a significant improvement in the efficiency for pre-forming molecules. One needs to be able to get the temperature low enough that the enhancement of the efficiency can be seen. This is a challenge for the experimentalists, but improvements of the efficiency are certainly likely to be seen if the experiments move onto optical lattices.

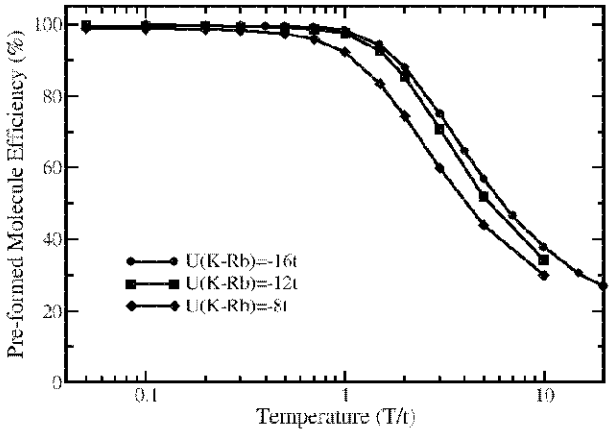

Figure 3. Efficiency versus temperature for the $20E_R$ lattice depth case and three different values of U_{bf} (−16t, −12t, −8t). Note how the efficiency rapidly rises to near 100% as the temperature is lowered and how there is relatively weak dependence on the K-Rb attraction.

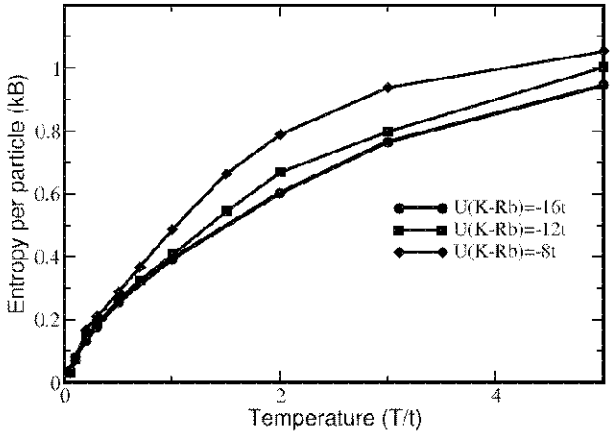

Figure 4. Entropy per particle versus temperature for the $20E_R$ lattice depth case and three different attractive interactions U_{bf} = −16t, −12t, and −8t

In Figure 4, we show a plot of the entropy per particle versus temperature for the same case. The entropy is an important quantity in cold atom experiments, because the trapped systems are isolated, and hence if modifications are made slow enough, the system remains adiabatic, and preserves its total entropy. Hence, one can use the entropy as a type of thermometer for the system that determines how low in temperature one can go on the lattice. The entropy here is high for the disordered high temperature phases, and then becomes small at low temperature, decreasing linearly at the lowest temperatures, as expected for this system. One can see that while there is some difference in entropy at a given temperature for the different attractions, once the temperature is low enough all of the curves coalesce to very similar values. In this regime, where the temperature is smaller than about half the bandwidth, the entropy is low, on the order of k_B or less. These entropy values will be challenging to achieve in the experiments, but are the realm one needs to get to in order to see the large enhancements to the efficiency. We feel that this is one of the most promising methods for rapidly improving the efficiency of the molecule formation using existing technology. The hardest aspect for this approach is the cooling problem.

Further results will be presented elsewhere, as will a comprehensive discussion of the science behind pre-formed molecules and the advantages one can find by employing optical lattices to improve the efficiency of dipolar molecule formation[14].

5. Conclusions

In this work, we have shown how to implement an efficient algorithm for the IDMFT approach to many-body physics. The algorithm has been applied to the problem of understanding how to improve the efficiency of dipolar molecule formation in optical lattices, which will set the stage for CAP Phase II work in the optical lattice emulator program currently being developed with support of a DARPA program and Army Research Office (ARO). We ran most of the code on the Cray XT5 machine at ARSC. We found the code scales well to large number of processors and operates at almost 100% of the peak speed for one operation per clock cycle. The scaling was nearly linear on up to 3,400 CPUs. On the science side, we found a significant improvement in the formation efficiency for molecules if one can achieve a low-enough temperature. By properly tuning the K-Rb attraction, one should be able to achieve nearly 100% conversion of the mixture into dipolar ground-state molecules.

Acknowledgments

This work was supported under ARO Award W911NF0710576 with funds from the DARPA OLE Program. We would also like to acknowledge collaborators on this project, which include Tom Hanna, Anzi Hu, Paul Julliene, Romuald Lemanski, Maciek Maska, and Carl Williams. Supercomputer time was provided under a CAP Phase II on the Cray XT5 at ARSC and from a Challenge Project.

References

1. Ni, K.-K., S. Ospelkaus, M.H.G. de Miranda, A. Pe'er, B. Neyenhuis, J.J. Zirbel, S. Kotochigova, P.S. Julienne, D.S. Jin, and J. Ye, "A High Phase-Space-Density Gas of Polar Molecules." *Science*, 322, pp. 231– 235, 2008.

2. Feynman, R.P., "Simulating physics with computers." *Int. J. Theor. Phys.*, 21, pp. 467–488, 1982.

3. Hubbard, J., "Electron Correlations in Narrow Energy Bands." *Proc. R. Soc. London. Ser. A, Mathematical and Physical Sciences*, 276, pp. 238–257, 1963.

4. Falicov, L.M. and J. C. Kimball, "Simple model for semiconductor-metal transitions: SmB_6 and transition-metal oxides." *Phys. Rev. Lett.*, 22, pp. 997–999, 1969.

5. Maska, M.M., R. Lemański, J.K. Freericks, and C.J. Williams, "Pattern formation in mixtures of ultracold atoms in optical lattices." *Phys. Rev. Lett.*, 101, 060404-1–4, 2008.

6. Freericks, J.K., and V. Zlatić, "Exact dynamical mean field theory of the Falicov-Kimball model." *Rev. Mod. Phys.*, 75, pp. 1333–1382, 2003.

7. Potthoff, M. and W. Nolting, "Metallic surface of a Mott insulator–Mott insulating surface of a metal." *Phys. Rev. B*, 60, pp. 7834–7849, 1999.

8. Freericks, J.K., *Transport in multilayered nanostructures: the dynamical mean-field theory approach*, Imperial College Press, London, 2006.

9. Tran, M.-T., "Inhomogeneous phases in the Falicov-Kimball model: Dynamical mean-field approximation." *Phys. Rev. B*, 73, 205110, 2006.

10. Helmes, R.W., A. Costi, and A. Rosch, "Mott Transition of Fermionic Atoms in a Three-Dimensional Optical Trap." *Phys. Rev. Lett.*, 100, 056403, 2008.

11. Snoek, M., I. Titvinidze, C. Toke, K. Byczuk, and W. Hofstetter, "Antiferromagnetic Order of Strongly Interacting Fermions in a Trap: Real-Space Dynamical Mean-Field Analysis." *New J. Phys.*, 10, 093008-1–9, 2008.

12. Brandt, U. and C. Mielsch, "Thermodynamics and correlation functions of the Falicov-Kimball model in large dimensions." *Z. Phys. B–Condens. Mat.*, 75, pp. 365–370, 1989; "Thermodynamics of the Falicov-Kimball model in large dimensions II." *Z. Phys. B–Condens.*, Mat. 79, pp. 295–299, 1990.

13. Freericks, J.K. and V.M. Turkowski, *Phys. Rev. B*, 2009 (to appear).

14. Freericks, J.K., M.M. Maska, Anzi Hu, T. Hanna, C.J. Williams, P. Julienne, and R. Lemanski, "Improving the efficiency of ultracold dipolar molecule formation by first loading onto an optical lattice." arxiv:0908.1794.