# Modeling Mixtures of Different Mass Ultracold Atoms in Optical Lattices: An Illustration of High Efficiency and Linear Scaling on the Cray XT4 via a Capability Applications Project at ERDC

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

## Abstract

*Defense Advanced Research Projects Agency (DARPA) is running a program to create an analog quantum-mechanical simulator for strongly interacting quantum particles. It is called an optical lattice emulator, and it involves ultracold neutral atoms moving in a corrugated periodic potential created by an optical lattice. The atoms interact with each other via collisions. In this work, we show how one can apply inhomogenenous dynamical mean-field theory (IDMFT) as an approximate computational tool to study the behavior of just such a system. Ultimately, conventional computation will be employed to benchmark the optical lattice emulator on a numerically tractable problem (a verification and validation study), and then the emulator will be applied to problems that cannot be solved with conventional computation. Here, we demonstrate an efficient massively parallel implementation of the IDMFT algorithm that is transportable, scales to many thousands of processors, and runs at approximately 50% of the theoretical peak speed of the machine.*

## 1. Introduction

In 1982, Richard Feynman proposed the idea of an analog quantum computer that could be used to simulate strongly interacting systems.[1] One simply creates an artificial system that behaves as the one that you wish to study, and then you let it evolve on its own over time and finally readout its properties. Carrying out such a program has proved to be difficult, but recently there has been a significant advance in achieving this goal, due to new experiments in ultracold atoms placed in optical lattices.

Using laser cooling and evaporation techniques, a number of different atomic systems can be made into the coldest "material" in the universe (at temperatures on the order of nano Kelvins). The atoms in these systems, which are typically the alkali atoms, interact with each other via collisions with scattering lengths that can often be tuned by something called a Feshbach resonance. In addition, by focusing retroreflected laser light onto the atomic cloud, one can create a standing wave of light that acts like a periodic potential for the atoms. The atoms are also trapped by an additional (often harmonic) trap that keeps them localized in a particular region of space. Atoms in such an optical lattice can move (via quantum-mechanical tunneling) between neighboring sites, and they interact with other atoms when two (or more) sit on the same lattice site. The quantum statistics of the particles (fermions or bosons) can be controlled by choosing an appropriate isotope for the different atomic species (although some atoms like Rb and Cs only have stable bosonic species).

The physical system, now consisting of a set of quantum particles 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[2] or the Falicov-Kimball model[3]). The atoms act like electrons (or effective bosonic particles), but now move on a much larger sized system, 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. Finally, since these systems have no defects and the lattices are essentially rigid, one can study the pure analog of correlated electrons without worrying about the purity of the system or the vibrations of the lattice. This cannot be done in solid state systems.

In our work, we focus on mixtures of two different mass atoms, both of them (spin-polarized) fermions. This could be made from mixtures of Li and K or from mixtures of Li or K with Sr or Yb. The system is put into an optical lattice, where the hopping of the light atom from site to site is large, but the hopping of the heavy atom from site to site is so small, it can be neglected.

978-0-7695-3515-9/08 $25.00 © 2008 IEEE
DOI 10.1109/DoD.HPCMP.UGC.2008.38

This means we ignore the quantum-mechanical effects of the heavy atom kinetic energy. But, we assume that the system can explore all possible positions for the heavy atoms, 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[3] which has two kinds of particles: itinerant particles with creation and annihilation operators $c_i^\dagger$ and $c_i$ for the light atoms at site $i$ (located at position $\mathbf{R}_i$) and localized particless with the corresponding operators $f_i^\dagger$ and $f_i$. The FK Hamiltonian is

$$\begin{aligned}\mathcal{H} = & -\sum_{ij} t_{ij} c_i^\dagger c_j + U \sum_i c_i^\dagger c_i f_i^\dagger f_i \\ & + \sum_i \left(V_i - \mu\right) c_i^\dagger c_i + \sum_i \left(V_i^f + E_f\right) f_i^\dagger f_i,\end{aligned} \quad (1)$$

where $-t_{ij}=-t$ is the nearest-neighbor hopping matrix, $U$ is the on-site repulsion between $c$ and $f$ particles, $\mu$ is the chemical potential of the mobile particles and $E_f$ is the local site energy of the localized particles. The two potentials $V_i$ and $V_i^f$ represent the additional trap, which is chosen to be harmonic for both species but with different tunable curvatures. We parametrize the traps with an effective length $R$ and $R^f$ where $V_i = tR_i^2/\left(2R^2\right)$ and $V_i^f = tR_i^2/\left(2R^{f2}\right)$.

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 $\mu$ and $E^f$ must be adjusted to produce the correct filling at each temperature for which the simulation is run. We fix $R_f$=30 and adjust $R$ to five different values 12.9, 17, 18.5, 20, and 30. We also fix $U$=5. In this case, the system is in a regime that favors phase separation of the two species at low temperature, but by squeezing the light atoms in a tighter trap, we can examine the crossover from when the heavies are on the inside and the lights on the outside to the opposite case with the lights on the inside and the heavies on the outside. These are precisely the parameters that have already been examined with quantum Monte Carlo simulations and with the local density approximation at $T$=0.[4] Here, we use inhomogenenous dynamical mean-field theory (IDMFT), because it allows us to examine many more temperatures, with much less critical solwing down in the simulation. IDMFT allows us to calculate the entropy, which is difficult to do for the quantum Monte Carlo simulations.

## 2. Algorithm

The IDMFT algorithm is based on the generalization of the highly successful dynamical mean-field theory[5] to inhomogeneous systems. This was first done for multilayered systems by Potthoff and Nolting[6] and now is the subject of a book[7]. More recently, Tran applied it to ultracold atoms in traps[8], and since then much work has followed[9,10].

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

$$G_{ij}\left(\tau\right) = -\mathrm{Tr}\, e^{-\beta\mathcal{H}} \mathcal{T}_\tau c_i\left(\tau\right) c_j^\dagger\left(0\right) \frac{1}{\mathcal{Z}}, \quad (2)$$

where Tr denotes the trace over all many-body eigenstates, $\beta$=$1/T$ is the inverse temperature, $\mathcal{T}_\tau$ is the time-ordering operator, which moves earlier times to the right (with a sign change if two fermionic operators are interchanged), and $\mathcal{Z}$=Tr exp[$-\beta\mathcal{H}$] is the partition function. The time-dependent fermionic creation and annihilation operators are written in the Heisenberg picture where

$$c_i\left(\tau\right) = e^{\tau\mathcal{H}} c_i e^{-\tau\mathcal{H}} \text{ and } c_i^\dagger\left(\tau\right) = e^{\tau\mathcal{H}} c_i^\dagger e^{-\tau\mathcal{H}}. \quad (3)$$

By using the invariance of the trace, one can show that the Green's function is antiperiodic in $\tau$ over the range $0\leq\tau\leq\beta$, 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. So we have

$$G_{ij}\left(i\omega_n\right) = \int_0^\beta d\tau e^{i\omega_n\tau} G_{ij}\left(\tau\right). \quad (4)$$

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

$$\sum_k \left[\left\{i\omega_n + \mu - V_i - \Sigma_i\left(i\omega_n\right)\right\}\delta_{ik} + t_{ik}\right] G_{kj}\left(i\omega_n\right) = \delta_{ij}, \quad (5)$$

so the Green's function is found by inverting the matrix defined in the square brackets. 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. The self-energy is calculated by solving an effective single-site impurity problem in a time-dependent field, which is determined self-consistently. Without going into details, the set of equations that the self-energy satisfies for the FK model in References 11 and 5

$$G_{ii}^0\left(i\omega_n\right) = \frac{1}{G^{-1}\left(i\omega_n\right) + \Sigma_i\left(i\omega_n\right)}, \quad (6)$$

$$G_{ii}\left(i\omega_n\right) = \left(1 - n_i^f\right) G_{ii}^0\left(i\omega_n\right) n_i^f \frac{1}{\left[G_{ii}^0\left(i\omega_n\right)\right]^{-1} - U}, \quad (7)$$

and

$$\Sigma_i\left(i\omega_n\right) = \left[G_{ii}^0\left(i\omega_n\right)\right]^{-1} - G_{ii}^{-1}\left(i\omega_n\right). \tag{8}$$

Here, the symbol $G_{ii}^0\left(i\omega_n\right)$ is called the effective medium and $n_i^f$ is the density of heavy particles 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 Eq. 5 at each lattice site *i*; (iii) determine the effective medium using the local Green's function and the old self-energy in Eq. 6; (iv) find the new Green's function from Eq. 7; and (v) find the new self-energy from Eq. 8 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.

In order to carry out the calculation, we still need to determine $n_i^f$ at each lattice site. The heavy particle filling is a functional of the Green's functions, and is expressed as $n_i^f = \mathcal{Z}_{1i}/\left(\mathcal{Z}_{0i} + \mathcal{Z}_{1i}\right)$ with

$$\mathcal{Z}_{0i} = 2e^{\beta(\mu - V_i)/2}\prod_{n=-\infty}^{\infty}\frac{1}{G_{ii}^0\left(i\omega_n\right)i\omega_n}, \tag{9}$$

and

$$\mathcal{Z}_{1i} = 2e^{\beta\left(E_f + V_i^f\right) + \beta\left(\mu - V_i - U\right)/2}\prod_{n=-\infty}^{\infty}\frac{\left[G_{ii}^0\left(i\omega_n\right)\right]^{-1} - U}{i\omega_n}. \tag{10}$$

Finally, the light particle filling is found from

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

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

Since we want to work with a fixed number of heavy and light particles, we need to run the calculation for a few different values of the chemical potentials $\mu$ and $E_f$, 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. But at low-temperature, when the system is phase separated, it sometimes is difficult to find a solution, because the filling can vary exceedingly rapidly with changes in the chemical potentials, making it challenging to achieve the target densities.

This IDMFT algorithm is well-suited for parallel implementation within a 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 heavy particle density, 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 matrix inversion for each Matsubara frequency to a different slave node. Since in both cases, the communications involves just vectors, rather than matrices, the code rarely encounters communications-based limitations in scaling.

The parallel implementation for the 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. 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.[12] We typically use between 64 ($T$=0.1) and 1,020 ($T$=0.05) positive Matsubara frequencies for a given calculation. Details for how such a scheme is implemented will appear elsewhere.[12]

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

$$\begin{aligned} s_i = -\int d\omega\rho_i\left(\omega\right)\Big[f\left(\omega\right)\ln f\left(\omega\right) \\ + \left\{1 - f\left(\omega\right)\right\}\ln\left\{1 - f\left(\omega\right)\right\}\Big] \\ -n_i^f\ln n_i^f - \left(1 - n_i^f\right)\ln\left(1 - n_i^f\right), \end{aligned} \tag{12}$$

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

The real-axis Green's function is found from analagous equations to those used for the imaginary axis, except now we know what the heavy particle 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 $\omega$ instead of a Matsubara frequency. Since we have a fixed grid of frequencies, the computational size is identical for all temperatures. We typically use 1,204 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 a sum-rule violation for the Green's functions if the trap is too tight, because we do not have frequency points at high enough frequencies to include all of the nonzero spectral weight of the Green's functions for the outermost lattice sites. Both of these issues are not serious and are well controlled in our computations.

## 3. Results

When the XT4 was originally configured at ERDC, it used dual core chips running at 1.8 GHz. There were approximately 2,100 boards or 4,200 processors available. Since each processor is capable of achieving two double precision arithmetic computations during each clock cycle, one could, in principle, run at speeds up to 3.6 Gflops per processor. The initial Capabilities Applications Project (CAP) was run on this configuration of Jade. After the Phase I of the CAP ended (scaling demonstration), the system went down for about six weeks and all boards were updated to 2.1 GHz quad cores, making approximately 8,400 CPUs available for computation (with a maximum flop rate of 4.2 Gflops). We performed a few scaling studies during Phase II of the CAP, but found that our results were similar enough to the phase I studies that they did not warrant further examination, and instead we focused on our production runs.

The IDMFT algorithm uses two main codes. The first, an imaginary axis code, determines the chemical potentials and then the local densities of the light and heavy atoms. 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, and since more computational time is spent on the imaginary axis code, most of the scaling and performance analysis was performed on that code.

We used a few different techniques to determine the scaling and performance of the codes. The simplest technique we used was just timings 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.

We went further than just a scaling analysis though. It is possible to have a code that is inefficient, but scales well, 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. This was done primarily for the strong-scaling case, where one expects there to be a degradation of the overall performance, as the computational speed is increased (due to more communications, etc.).

Finally, we tested two different implementations of LAPACK and BLAS on Jade. We examined the xt-LibSci implementation and we examined the ACML implementation. The Portland group FORTRAN77 compiler was used and the compiler flags were set to -fast. We found no significant improvement in performance with any other flag options for the compiler.

In Figure 1, we show the strong scaling analysis for the imaginary axis code. The theoretical maximum speed up (linear curve) is compared to the dual core (black line with circles) and quad core (red line with squares) One can immediately see that the code is giving almost perfect linear scaling up to 4,096 CPUs. The dual core case is better than 90% of linear scaling, while the quad core case is closer to 99% of linear scaling. This is outstanding performance for strong scaling.

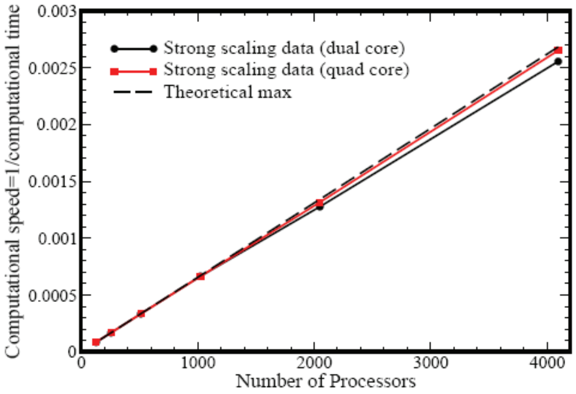

**Figure 1. Strong-scaling data for the imaginary axis code on both the dual core and quad core versions of Jade. The theoretical maximum (dashed line) is found by fitting the computational speed versus number of CPUs for small CPUs and extrapolating the linear curve. Note how the dual core configuration had better than 90% scaling, while the quad core is coming in at more than 99% of strong scaling. Such performance is outstanding.**

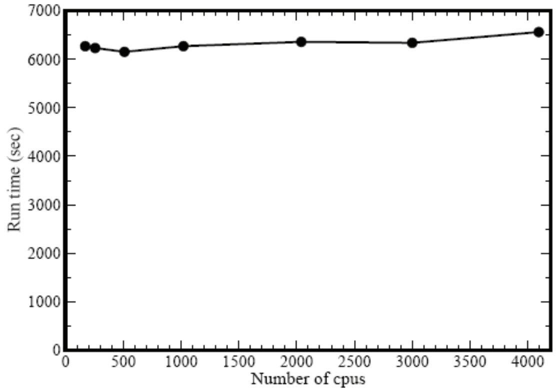

**Figure 2. Weak-scaling data for the imaginary axis code on the dual core version of Jade. Note how the weak scaling curve has approximately the same computational time regardless of the size of the job.**

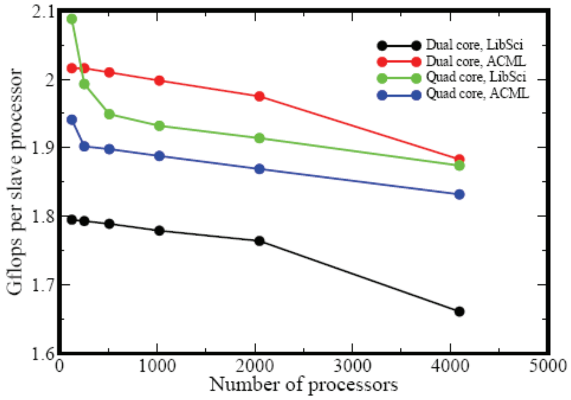

**Figure 3. Performance analysis of the imaginary axis code using the PAPI suite. Two curves are shown for each configuration of the machine (dual core and quad core). One uses the LibSci math library and the other uses the ACML math library. Note the greater variation in the flop rates for the quad core machines.**

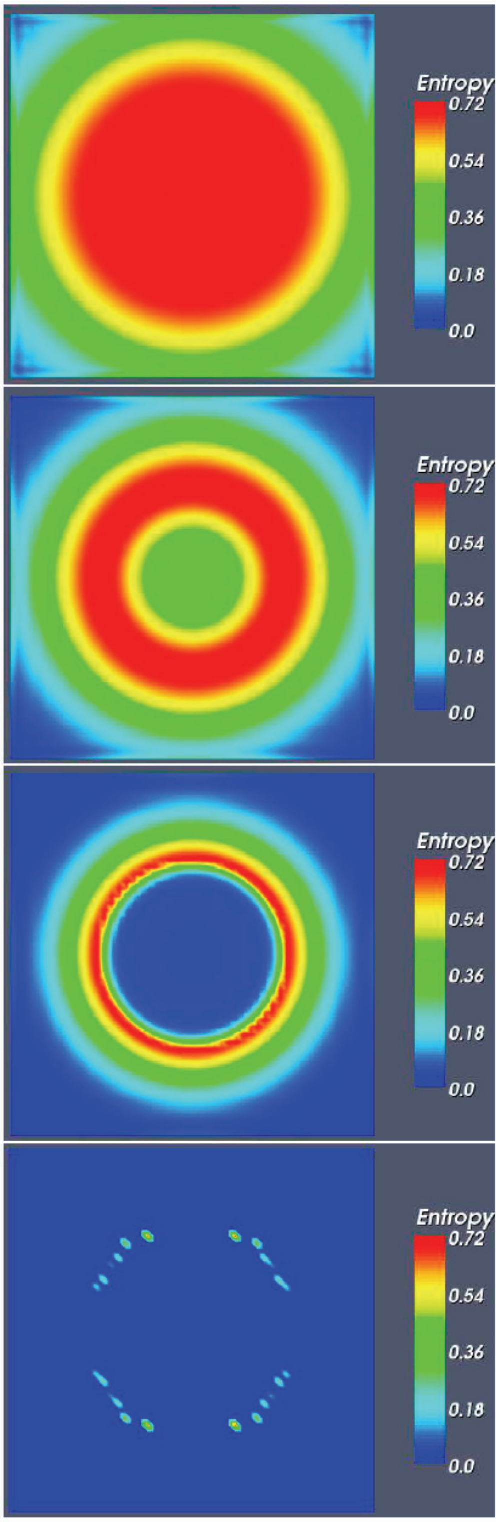

**Figure 4. False color plot of the entropy distribution for the case with *R*=30, *U* =5, and various temperatures (top to bottom: *T*=0.2, *T*=0.125, *T*=0.075, *T*=0.01). Ordering begins at *T* ≈ 0.125, and the entropy is quite small at the lowest *T*.**

Next, we examine the weak scaling in Figure 2. Note how there are some small variations, but in general, the weak scaling curve is showing a nearly constant run time as the job size and number of CPUs are increased.

Now we move on to the performance analysis using Performance Application Programming Interface (PAPI). We ran the performance analysis for the strong scaling cases already shown in Figure 1, but we also included an analysis of the different package options for the LAPACK and BLAS routines. Our two choices were the LibSci library and the ACML library. The results are shown in Figure 3, and are strange. To begin, we are achieving something in the vicinity of 50% of peak speed on each CPU (dual core or quad core) with the percentage being slightly higher for the dual core case. The drop off in performance is quite moderate for the dual core case, and is a bit more rapid for the quad core case. Surprisingly, in the dual core case, the ACML library was faster, while in the quad core case it is the LibSci library that is faster. Even more surprising is the fact that on the quad cores we see a definite drop in the flop rate as we move to larger and larger numbers of CPUs, but our scaling analysis based on the total run time is showing better linear scaling for the quad cores. It is hard to reconcile these facts. One possibility is that the implementation of PAPI on the quad core Jade is having errors in accurately measuring the number of floating point operations on each CPU, and the flop rates are not so accurate. This issue is one we have not been able to resolve.

We briefly discuss some of the scientific results from this work. In Figure 4, the entropy distribution is plotted for the case with $R$=30 and $U$=5. The panels are for different temperatures ranging from hot at the top to cold at the bottom. Note how the entropy, which is initially distributed primarily in the center of the trap, moves to an annulus and eventually becomes very small at the lowest temperature. Even at $T$=0.01, the dominant contribution to the entropy is coming from the heavy particles in the regions where the heavy particle density is non-integer on particular lattice sites; this is the boundary region of the phase separation. This can be seen more clearly in Figure 5, where density plots for the heavy particles are shown for the same four temperatures. Here we can now clearly see that ordering starts near $T\approx0.125$, and becomes nearly complete at the lowest $T$.

Further results will be presented elsewhere, when the work is complete.

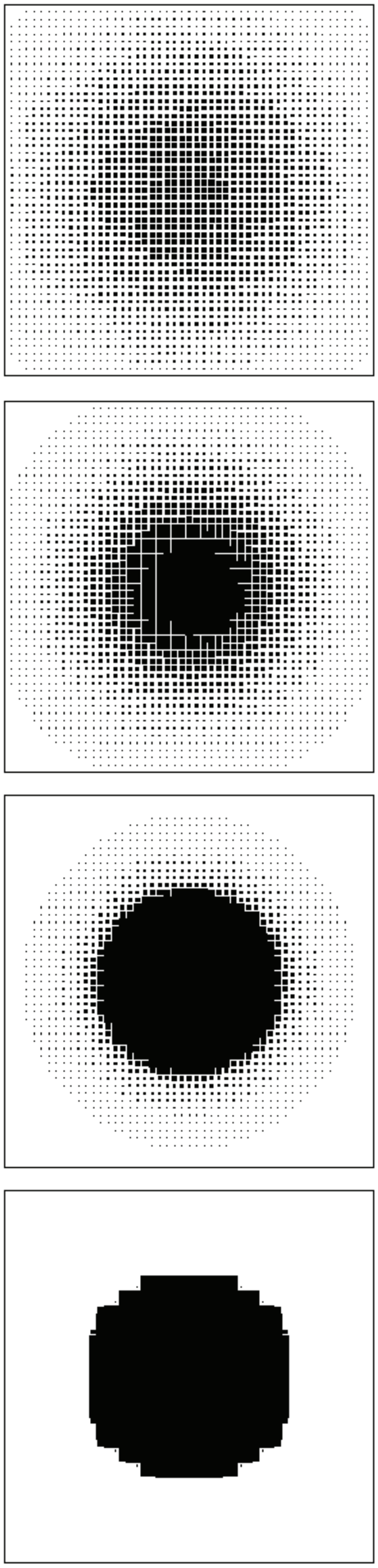

**Figure 5. Density distribution of the heavy atoms for the case with $R$=30, $U$ =5, and various temperatures (top to bottom: $T$=0.2, $T$=0.125, $T$=0.075, $T$=0.01). Ordering begins at $T\approx0.125$, and is nearly complete at the lowest $T$.**

## 4. 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 the behavior of mixtures of different mass atoms in optical lattices, which will serve as one potential benchmark calculation for the optical lattice emulator currently being developed with the support of a Defense Advanced Research Projects Agency (DARPA) program. We ran most of the code on the Cray XT4 machine at the US Engineer Research and Development Center (ERDC), in both its dual core and quad core configurations. We found the code scales well to a large number of processors and operates at almost 50% of the theoretical peak speed on the largest size we ran on, which was 4,096 CPUs.

## Acknowledgments

This work was supported under Army Research Office Award W911NF0710576 with funds from the DARPA OLE Program. We would also like to acknowledge useful conversations with T-L. Ho, H.R. Krishnamurthy, and C.J. Williams. J. Thomas provided technical assistance with the implementation of PAPI. Supercomputer time was provided under CAP Phase II on the Cray XT4 at ERDC.

## References

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

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

3. 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.

4. Maska, M.M., R. Lemański, J.K. Freericks, and C.J. Williams, "Pattern formation in mixtures of ultracold atoms in optical lattices." *arXiv:0802.3894* (preprint), 2008.

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

6. 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.

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

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

9. 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.

10. 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." *arXiv:0802.3211* (preprint), 2008.

11. 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.

12. Freericks, J.K. and V.M. Turkowski, *unpublished*.