Philosophical Magazine

ISSN: 1478-6435 (Print) 1478-6443 (Online) Journal homepage: http://www.tandfonline.com/loi/tphm20

Recent advances in determinant quantum Monte Carlo

Chia-Chen Chang, Sergiy Gogolenko, Jeffrey Perez, Zhaojun Bai & Richard T. Scalettar

To cite this article: Chia-Chen Chang, Sergiy Gogolenko, Jeffrey Perez, Zhaojun Bai & Richard T. Scalettar (2015) Recent advances in determinant quantum Monte Carlo, Philosophical Magazine, 95:12, 1260-1281, DOI: 10.1080/14786435.2013.845314

To link to this article: http://dx.doi.org/10.1080/14786435.2013.845314

Published online: 10 Oct 2013.

Submit your article to this journal

Article views: 469

View related articles

View Crossmark data

Citing articles: 2 View citing articles

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=tphm20

Download by: [University of California Davis] **Date:** 27 April 2016, At: 09:40

Philosophical Magazine, 2015
Vol. 95, No. 12, 1260–1281, http://dx.doi.org/10.1080/14786435.2013.845314

Taylor & Francis
Taylor & Francis Group

Recent advances in determinant quantum Monte Carlo

Chia-Chen Chang[a], Sergiy Gogolenko[b], Jeffrey Perez[b], Zhaojun Bai[b] and Richard T. Scalettar[a]*

[a]*Department of Physics, University of California, Davis, California 95616, USA;* [b]*Department of Computer Science, University of California, Davis, California 95616, USA*

(*Received 28 May 2013; accepted 7 September 2013*)

Determinant quantum Monte Carlo is a method for studying magnetic, transport and thermodynamic properties of interacting fermions on a lattice. It is widely used to explore the physics of strongly correlated quantum systems, from cuprate superconductors to ultracold atoms trapped on optical lattices. This paper contains a description of recent algorithmic advances in the determinant quantum Monte Carlo technique. Focus will be on algorithms developed for hybrid multicore processor and GPU platforms. The resulting speed-up of the simulations will be quantified. Simulations' results will also be presented, with an emphasis on physical quantities that can now be computed for large numbers of sites.

Keywords: Hubbard model; Mott insulators; Monte-Carlo; multilayers

Downloaded by [University of California Davis] at 09:40 27 April 2016

1. Introduction

One of the most prominent areas of theoretical condensed matter physics is the study of 'model Hamiltonians' which, without incorporating the detailed chemical realism of a specific material, nevertheless attempt to describe fundamental properties like magnetism and metal-insulator transitions of a collection of quantum particles moving on a lattice of sites. Such models, because of their greater simplicity compared to the solution of the many-body Schroedinger equation in the presence of a collection of nuclei, allow for more accurate treatments of the effects of the interactions between the particles, which otherwise might need to be handled in a more approximate (e.g. mean field) fashion.

Determinant quantum Monte Carlo (DQMC) [1] is one of the methods which can be applied to such models. DQMC has a number of advantages relative to other approaches: it solves the interacting many body problem exactly and can compute a broad range of response functions, such as magnetic susceptibility and compressibility. It also has several weaknesses. One of the most significant is that the method scales as the cube of the number of lattice sites, and hence at fixed density as the cube of the number of fermions. Although much algorithmic progress has been made, simulations of several hundred to a thousand sites remain at the upper end of feasibility. This is an important limitation in a variety of situations. First, if one wants to explore the occurrence of spontaneous symmetry breaking to an ordered low temperature (e.g. magnetic or superconducting) phase, it is necessary to extrapolate in

*Corresponding author. Email: scalettar@physics.ucdavis.edu

© 2013 Taylor & Francis

Downloaded by [University of California Davis] at 09:40 27 April 2016

system size. In 2D, lattice sizes like 8×8, 16×16, 24×24 and 32×32 allow one to get a pretty good sense of finite size effects; but in 3D, $4\times4\times4$, $6\times6\times6$, $8\times8\times8$ and $10\times10\times10$ lattices have much smaller linear extent. Second, to look at inhomogeneous systems, large lattices are required to capture the variation of behaviour across the lattice. There are many interesting systems for which the examination of inhomogeneities is crucial, including, for example, stripe phases of cuprate superconductors [2–4] and confined, ultracold gases on optical lattices [5].

DQMC is of course not the only approach to these problems. Methods like dynamical mean-field theory (DMFT) [6–8] and its cluster extensions [9] complement DQMC. They work in the thermodynamic limit, and hence can capture divergences which occur at phase transitions. However, they (especially DMFT) make other sorts of approximations. Diagrammatic QMC [10] is a method which currently looks at lattices of roughly the same sizes as DQMC. Like the advances we will describe here, rapid progress is being made in these approaches.

The purpose of this paper is to describe our recent efforts on pushing DQMC to simulate larger lattice sizes (i.e. speeding up the algorithm) by means of taking advantage of modern multicore CPU and GPU hardware. These developments do not alter the basic formalism of DQMC, which we summarize in Section II. Rather, focus will be on making the linear algebra kernels, which are at the heart of DQMC, more efficient. An important consideration will be on ensuring that numerical stability is maintained in the course of these modifications [11–14].

Like many Monte Carlo-based simulation methods for fermions, DQMC suffers from the so-called 'sign problem',[15–17] which occurs when the probability distribution used to sample configurations becomes negative. We do not address the sign problem in this paper. Solving the sign problem remains the central issue in the field of QMC simulation for fermions.

2. Formalism of DQMC

In this section, we present the essential ingredients of the determinant quantum Monte Carlo technique. We refer the readers to references [1,13,18] for more detailed discussions of the method.

2.1. *The Hubbard model*

DQMC is typically formulated for lattice models. One of the simplest (and most important) is the Hubbard Hamiltonian for electrons,

$$\hat{H} = \underbrace{-t \sum_{\langle ij\rangle\sigma} (\hat{c}^{\dagger}_{i\sigma}\hat{c}_{j\sigma} + \hat{c}^{\dagger}_{j\sigma}\hat{c}_{i\sigma}) - \sum_{i} \mu_i(\hat{n}_{i\uparrow} + \hat{n}_{i\downarrow})}_{\hat{T}} + \underbrace{U \sum_{i} \hat{n}_{i\uparrow}\hat{n}_{i\downarrow}}_{\hat{V}}. \tag{1}$$

The operator $\hat{c}^{\dagger}_{i\sigma}(\hat{c}_{i\sigma})$ creates (annihilates) an electron of spin σ ($\sigma = \uparrow, \downarrow$) on site i. $\hat{n}_{i\sigma} = \hat{c}^{\dagger}_{i\sigma}\hat{c}_{i\sigma}$ is the number operator for spin-σ particles. $\langle \ldots \rangle$ denotes that the summation

Downloaded by [University of California Davis] at 09:40 27 April 2016

is over nearest-neighbour sites. $\hat{T}$ represents collectively the terms that are quadratic in creation/annihilation operators and $\hat{V}$ denotes the interaction term. The parameters t and U control the kinetic and interaction energies, respectively, and the chemical potential μ_i is used to adjust the density. The Hubbard Hamiltonian assumes that, due to screening effect, the interaction between electrons is local, i.e. only electrons of opposite spins on the same lattice site can feel the interaction. In most cases, t is used as the scale of energy and one sets $t = 1$. The interaction strength U typically lies in the range $2 \leq |U| \leq 16$.

Although one normally describes i, j in Equation (1) as site indices, they can be more generally regarded as site and/or orbital labels. Thus, the parameter t can be extended to describe both inter-site and inter-orbital hoppings. Likewise, the chemical potential μ_i might then reflect different orbital energies.

The electron creation and annihilation operators act on a Hilbert space spanned by a set of occupation number basis. A given basis specifies the number of spin-↑ and spin-↓ electrons $(n_{i\uparrow}, n_{i\downarrow})$ for each lattice site or orbital. Because of the Pauli exclusion principle, each site/orbital is allowed to have 4 number configurations: $(n_{i\uparrow}, n_{i\downarrow}) = (0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$. Therefore, the dimension of the Hilbert space is 4^N, where N is the number of sites/orbitals. Thermodynamic properties of the Hubbard Hamiltonian can be obtained through computing the expectation value of an operator $\hat{O}$:

$$\langle \hat{O} \rangle = \frac{1}{Z} \operatorname{Tr} \left(e^{-\beta \hat{H}} \hat{O} \right), \qquad Z = \operatorname{Tr} e^{-\beta \hat{H}}. \tag{2}$$

Tr denotes the trace over the 4^N-dimensional Hilbert space. $\beta = 1/T$ is the inverse of temperature T. Z is the so-called partition function. Temperatures of interest depend strongly on the physics being examined. Local observables like the energy and near neighbour correlation functions typically require $T \sim t/3$ (i.e. $\beta t \sim 3$), while the observation of long range order across a large lattice might only develop at $T \sim t/20$ (i.e. $\beta \sim 20$). As we shall discuss below, larger β simulations are more difficult and costly.

One straightforward way of computing $\langle \hat{O} \rangle$ is to first obtain the exact eigenstates of $\hat{H}$ and then carry out the trace explicitly. This amounts to diagonalizing a $4^N \times 4^N$ matrix. Clearly, this is feasible only for small lattices and current state-of-the-art exact diagonalization study of the Hubbard model is able to reach system size of the order $N \sim 20$. As we shall see in the following, the DQMC method reduces this exponential scaling to a power law, allowing one to treat systems as large as $N = 400 \sim 1000$.

2.2. *The DQMC method*

We begin the discussion with the partition function Z defined in Equation (2). A key element of the DQMC method is the recognition of the fact that for an operator in bilinear form of $\hat{c}_{i\sigma}$ and $\hat{c}^{\dagger}_{i\sigma}$:

$$\hat{A}_\sigma = \sum_{ij,\sigma} \hat{c}^{\dagger}_{i\sigma} A^{\sigma}_{ij} \hat{c}_{j\sigma} \equiv \mathbf{c}^{\dagger}_{\sigma} A_\sigma \mathbf{c}_\sigma, \tag{3}$$

the trace of $e^{-\hat{A}_\sigma}$ can be evaluated explicitly. In the above equation, $\mathbf{c}^{\dagger}_{\sigma} = (\hat{c}^{\dagger}_{1\sigma}, \hat{c}^{\dagger}_{2\sigma}, \ldots, \hat{c}^{\dagger}_{N\sigma})$. Specifically, we have

$$\operatorname{Tr} e^{-\hat{A}_\sigma} = \det \left(I + e^{-A_\sigma} \right), \tag{4}$$

where A_σ is the $N \times N$ matrix defined in Equation (3) and I is an identity matrix. Equation (4) can be readily generalized to multiple bilinear operators

$$\mathrm{Tr}\left(e^{-\hat{A}_{1\sigma}} e^{-\hat{A}_{2\sigma}} \dots e^{-\hat{A}_{m\sigma}}\right) = \det\left(I + e^{-A_{1\sigma}} e^{-A_{2\sigma}} \dots e^{-A_{m\sigma}}\right). \tag{5}$$

In the Hamiltonian Equation (1), the terms represented by $\hat{T}$ are in bilinear form, while the interaction term $\hat{V}$ is *quartic* in the electron operators $\hat{n}_{i\uparrow}\hat{n}_{i\downarrow} = \hat{c}^\dagger_{i\uparrow}\hat{c}_{i\downarrow}\hat{c}^\dagger_{i\uparrow}\hat{c}_{i\downarrow}$. In order to take advantage of Equation (5), the interaction term $\hat{V}$ will need to be casted into quadratic form. To proceed, we discretize β into many intervals $\beta = L\epsilon$, where L is an integer and ϵ is a real number. The exponential of $-\beta\hat{H}$ becomes

$$e^{-\beta\hat{H}} = \underbrace{e^{-\epsilon\hat{H}} e^{-\epsilon\hat{H}} \cdots e^{-\epsilon\hat{H}}}_{L \text{ terms}}. \tag{6}$$

For each $e^{-\epsilon\hat{H}}$, if ϵ is sufficiently small, one can invoke the Trotter-Suzuki decomposition [19–22] and write

$$e^{-\epsilon\hat{H}} \approx e^{-\epsilon\hat{V}} e^{-\epsilon\hat{T}}. \tag{7}$$

Because $\hat{T}$ and $\hat{V}$ do not commute, the decomposition is not exact and bears an error of order $\mathcal{O}(\epsilon^2)$. Nonetheless, the error can be systematically reduced by using higher order decompositions. Additionally, one can eliminate the error by carrying out simulations on a grid of different ϵ's and extrapolating the result to the $\epsilon \to 0$ limit.

The next step is to convert the exponent of $e^{-\epsilon\hat{V}}$ into bilinear form. This is achieved by implementing the discrete Hubbard-Stratonovich (HS) transformation [23]:

$$\begin{aligned} e^{-\epsilon U \sum_i n_{i\uparrow} n_{i\downarrow}} &= \frac{1}{2^N} e^{U\sum_i (n_{i\uparrow}+n_{i\downarrow})/2} \\ &\quad \times \sum_{s_{1\ell},\dots,s_{N\ell}=\pm 1} e^{\lambda \sum_i s_{i\ell}(n_{i\uparrow}-n_{i\downarrow})} \\ &= \frac{1}{2^N} \sum_{\vec{s}_\ell} \prod_\sigma e^{\mathbf{c}^\dagger_\sigma \mathcal{V}_{\sigma\ell} \mathbf{c}_\sigma}. \end{aligned} \tag{8}$$

Here, the parameter λ obeys $\cosh\lambda = e^{U\epsilon/2}$. We have attached site and time slice indices i and ℓ to the HS field $s_{i\ell}$ because this transformation is done for all sites/orbitals i and for each imaginary time slice $\ell = 1, 2, \cdots L$. $\vec{s}_\ell = (s_{1\ell}, s_{2\ell}, \dots, s_{N\ell})$ denotes collectively N Hubbard-Stratonovich fields at time slice ℓ. With the help from Equation (8), the Hamiltonian operator Equation (7) can be expressed as

$$e^{-\epsilon\hat{H}} = \sum_{\vec{s}_\ell} \prod_\sigma e^{\mathbf{c}^\dagger_\sigma \mathcal{V}_{\sigma\ell} \mathbf{c}_\sigma}\, e^{-\epsilon\, \mathbf{c}^\dagger_\sigma \mathcal{T}_\sigma \mathbf{c}_\sigma}, \tag{9}$$

where, following the notation in Equation (3), $\mathcal{T}_\sigma$ is the matrix representation of spin-σ part of the $\hat{T}$ operator[1].

At this point, the exponents of the righthand side of Equation (9) are all quadratic in electron operators. The trace can now be evaluated explicitly. To simplify the notation, let

Downloaded by [University of California Davis] at 09:40 27 April 2016

us define $\hat{B}_\ell^\sigma = e^{\mathbf{c}_\sigma^\dagger \mathcal{V}_{\sigma\ell} \mathbf{c}_\sigma}\, e^{-\epsilon\, \mathbf{c}_\sigma^\dagger \mathcal{T}_\sigma \mathbf{c}_\sigma}$. From Equation (5), the partition function Z can be expressed as[2]

$$\begin{aligned} Z &= \sum_{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_L} \prod_\sigma \mathrm{Tr}\left(\hat{B}_L^\sigma \hat{B}_{L-1}^\sigma \ldots \hat{B}_1^\sigma\right) \\ &= \sum_{\{\vec{s}\}} \det M_{\vec{s}}^\uparrow \, \det M_{\vec{s}}^\downarrow . \end{aligned} \tag{10}$$

$$M_{\vec{s}}^\sigma = I + B_L^\sigma B_{L-1}^\sigma \ldots B_1^\sigma . \tag{11}$$

In the above equations, $B_\ell^\sigma = e^{\mathcal{V}_{\sigma\ell}}\, e^{-\epsilon \mathcal{T}_\sigma}$ is an $N \times N$ matrix representing the operator $\hat{B}_\ell^\sigma$. The subscript $\vec{s}$ emphasizes that $M_{\vec{s}}^\sigma$ (and B_ℓ^σ) are functions of HS fields.

Using Equation (10), we can rewrite the expectation value of observable $\hat{O}$ as the following

$$\langle \hat{O} \rangle = \frac{1}{Z} \mathrm{Tr}\left(e^{-\beta \hat{H}}\, \hat{O}\right) = \sum_{\{\vec{s}\}} \mathcal{P}_{\vec{s}}\, \langle \hat{O} \rangle_{\vec{s}}, \tag{12}$$

where the probability density $\mathcal{P}_{\vec{s}}$ is defined as

$$\mathcal{P}_{\vec{s}} = \frac{1}{Z} \det M_{\vec{s}}^\uparrow \, \det M_{\vec{s}}^\downarrow , \tag{13}$$

and $\langle \hat{O} \rangle_{\vec{s}}$ is a local estimator of the observable $\hat{O}$ at a given field configuration $\{\vec{s}\}$:

$$\langle \hat{O} \rangle_{\vec{s}} = \frac{\mathrm{Tr}\left(e^{-\beta \hat{H}}\, \hat{O}\right)}{\det M_{\vec{s}}^\uparrow \, \det M_{\vec{s}}^\downarrow} . \tag{14}$$

Equations (12)–(14) are the central result of the DQMC method. The 4^N-dimensional trace has been converted into a summation over NL auxiliary fields. The sum is then carried out by the Metropolis Monte Carlo sampling algorithm, using $\mathcal{P}_{\vec{s}}$ as the probability distribution function. The precise sampling procedure is described in Refs. [1,13,18]. Here, we want to point out that the naive computational complexity of the Monte Carlo sampling is $\mathcal{O}(N^3)$ to update each variable s_{il}. However, this can be reduced to $\mathcal{O}(N^2)$ after implementing a technique which takes advantage of the special structure of the change to the matrices involved in the update.

A few qualitative comments are in order. First, the 'sign problem' originates in the fact that $\mathcal{P}_{\vec{s}}$ can become negative, so that its interpretation as a probability is compromised. There are several special situations when $\mathcal{P}_{\vec{s}}$ is positive, because the signs of the determinants of the two spin species are identical. This occurs, for example, in the attractive Hubbard model ($U < 0$) for all temperatures, hoppings and chemical potentials, and also for the repulsive Hubbard model at half filling on a bipartite lattice. Fortunately, there is much interesting physics in these cases. When the sign problem is present, the accessible temperatures depend on density ρ, interaction strength U and spatial lattice size N. Almost all knowledge of the value of the sign is empirical. Indeed, one of the complicated features of the sign problem is that it does not depend solely on the physical parameters of the model, but can vary according to the algorithmic details, such as the choice of Hubbard-Stratonovich transformation [24–26]. Some general patterns are that the sign becomes smaller as U is increased. This is not surprising since there is no sign problem in the non-interacting limit $U = 0$. The sign behaves better at densities ρ

Downloaded by [University of California Davis] at 09:40 27 April 2016

which correspond to 'filled shells' for a given lattice size, that is when the chemical potential is such that a collection of non-interacting energy levels $\epsilon(k)$ (are eigenvalues of the kinetic energy term in the Hamiltonian) has just been filled. The sign rapidly decreases to zero with increasing β once some negative configurations have begun to appear. If the average sign begins to deviate from unity at β_0, typically simulations are feasible only to a maximal $\beta \sim \beta_0 + 2/t$. Interestingly, the behaviour of the average sign with spatial lattice size N is much more benign. This is a bit surprising since initial understanding of the sign problem indicated that the average sign should scale to zero with the same exponential dependence on both N and β. Refs. [13,15,27] contain data for different lattice sizes, temperatures, fillings and interaction strengths for the 2D and 3D Hubbard models, respectively.

A second comment is that severe round-off errors can accumulate in the multiplication of the B_ℓ^σ matrices, especially for large U and low temperatures. This problem has essentially been solved by 'stabilization methods' which separate and prevent mixing of the different numerical scales in the linear algebra operations. (Again, for details, see Refs. [1,13,18].) However, when formulating any new linear algebra algorithm to take advantage of new hardwares, care must be taken so that stability is preserved. Indeed, this is a key consideration in our assessment of new pivoting strategies, as will be discussed below.

In principle, one can write down a HS identity similar to Equation (8) for inter-site U_{ij} and even Hund's rule J_{ij} terms which flip spins on different orbitals [24]. In practice, however, the sign problem is very bad when these terms are present, and simulations can typically not reach low enough temperatures to see much of the interesting strong correlation physics.

2.3. *Green's function definition*

In the study of interacting systems, the Green's function is perhaps one of the most important observables to be calculated. Using Equation (14), one can show that the equal-time Green's function is expressed in a particularly simple fashion:

$$G^\sigma_{\vec{s},ij} \equiv \langle \hat{c}_{i\sigma} \hat{c}^\dagger_{j\sigma} \rangle_{\vec{s}} = \left[\left(M^\sigma_{\vec{s}} \right)^{-1} \right]_{ij} \tag{15}$$

From DQMC's point of view, the importance of the Green's function is twofold. Firstly, since the Green's function is the inverse of $M^\sigma_{\vec{s}}$, it is directly tied to the Monte Carlo sampling procedure. Secondly, for a given HS field configuration, one can invoke the Wick's theorem and express any observable in terms of the Green's function. Equation (15) can be generalized to include imaginary time dependence. The resulting observable is called the time-dependent Green's function:

$$G^\sigma_{\vec{s},ij}(\ell_1, \ell_2) \equiv \begin{cases} \langle \hat{c}_{i\sigma}(\ell_1) \hat{c}^\dagger_{j\sigma}(\ell_2) \rangle_{\vec{s}} & \text{if } \ell_1 > \ell_2 \\ -\langle \hat{c}^\dagger_{j\sigma}(\ell_2) \hat{c}_{i\sigma}(\ell_1) \rangle_{\vec{s}} & \text{if } \ell_2 > \ell_1 \end{cases} \tag{16}$$

where $1 \leq \ell_1, \ell_2 \leq L$ are imaginary time slice indices.

Crucial to the algorithmic discussions in the next section is the observation that the determinant of the dense $N \times N$ matrix in Equation (11) can also be written as the determinant

Downloaded by [University of California Davis] at 09:40 27 April 2016

of a sparse, but larger matrix of dimension $(NL) \times (NL)$:

$$\mathcal{M}_{\vec{s}}^{\sigma} = \begin{pmatrix} I & 0 & 0 & \cdots & 0 & B_1^{\sigma} \\ -B_2^{\sigma} & I & 0 & \cdots & 0 & 0 \\ 0 & -B_3^{\sigma} & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -B_L^{\sigma} & I \end{pmatrix}_{\vec{s}}. \tag{17}$$

That is $\det \mathcal{M}_{\vec{s}}^{\sigma} = \det M_{\vec{s}}^{\sigma}$. Moreover, the inverse of Equation (17) generates all necessary Green's functions: the diagonal elements of $(\mathcal{M}_{\vec{s}}^{\sigma})^{-1}$ are equal-time Green's functions and the off-diagonal ones correspond to time-dependent Green's functions.

Downloaded by [University of California Davis] at 09:40 27 April 2016

3. Green's function calculation

The purpose of this section is to highlight recent algorithmic advances in the calculation of the equal-time and time-dependent Green's functions. The algorithm's performance benchmark on multicore platforms and multicores accelerated with GPUs will also be discussed. In the following discussions, for the sake of simplicity, the spin index σ and HS field $\vec{s}$ will be ignored.

3.1. *Equal-time Green's function*

The calculation of the equal-time Green's function involves a long chain of matrix multiplications, see Equation (11). These products tend to be extremely ill-conditioned for simulations with large repulsive interactions U and low temperatures T. Round-off errors in computing $B_L B_{L-1} \ldots B_1$, can be significantly reduced by using matrix chain multiplication approaches which stratify the magnitude of elements. The stratification process is aimed to protect small numbers from being rounded off by mixing them with large ones in matrix products.

The standard numerical tool for magnitude stratification is a QR decomposition with column pivoting (QRCP). The majority of traditional algorithms for computing G are based on graded decompositions generated by the QRCP. One of those algorithms was proposed by Loh et al. in Ref. [28]. The multiplication phase of Loh's approach consists of computing the chained product of B_ℓ in the factorized form $Q_L D_L T_L$ using QRCP. All matrix-matrix multiplications in this phase involve matrices that have elements of similar magnitude except for multiplications on diagonal matrices D_i which are equivalent to numerically stable column scaling operations. In order to improve accuracy in computing the inverse $(I + Q_L D_L T_L)^{-1}$, the method splits $D_L = D_b D_s$ into high magnitude D_b and low magnitude D_s terms in its inversion phase, where

$$D_b(i) = \begin{cases} 1/|D_L(i)|, & \text{if } |D_L(i)| > 1 \\ 1, & \text{otherwise} \end{cases},$$

$$D_s(i) = \begin{cases} D_L(i), & \text{if } |D_L(i)| \le 1 \\ \operatorname{sgn}(D_L(i)), & \text{otherwise} \end{cases}.$$

```
1: procedure EqualTimeG({B_i|1 ≤ i ≤ L})
2:     Compute QRCP: Q_1 R_1 P_1^T = B_1
3:     D_1 ← diag(R_1); T_1 ← D_1^{-1} R_1 P_1^T
4:     for i ∈ {2, 3, ..., L} do
5:         C_i ← (B_i Q_{i-1}) D_{i-1}
6:         Compute the permutation P_i: C_i P_i = C̃,
7:         Compute QR: C̃ = Q_i R_i
8:         D_i ← diag(R_i); T_i ← (D_i^{-1} R_i)(P_i^T T_{i-1})
9:     end for
10:    G ← (T_L^{-T} Q_L^T D_b^{-1} + D_s)^{-T} D_b^{-1} Q_L^T
11:    return G
12: end procedure
```

Figure 1. Loh's stratification method with pre-pivoting. Here parentheses reflect the order of operations, A^{-T} corresponds to $(A^T)^{-1}$, the function diag(A) returns diagonal matrix D with elements $(D_{ij})_{i,j=1}^{n} = (\delta_{ij} A_{ij})_{i,j=1}^{n}$, and δ_{ij} denotes Kronecker delta.

Downloaded by [University of California Davis] at 09:40 27 April 2016

The intensive use of QRCP is a performance bottleneck of Loh's stratification algorithm. The blocked QRCP algorithm implemented in LAPACK requires matrix-vector multiplications for preparing blocked matrix update, and thus have a low data reuse ratio. This imposes a limit to the performance on modern multicore architectures with deep memory hierarchies. A fast computable alternative to the QRCP was recently proposed in Ref. [29]. This novel approach is based on the observation that as Loh's algorithm iterates, it produces almost column graded matrices C_i, for which permutation matrices P_i specify only a few column interchanges from the initial ordering. As a result, the QRCP in the iterative steps of Loh's algorithm can be safely replaced by the QR with pre-pivoting, where a pre-pivoting permutation P_i is computed and applied, and then a regular QR is utilized. The permutation P_i for the pre-pivoting is based on sorting matrix columns by 2-norms in a descending order. The resulting algorithm for computing equal-time Green's function is illustrated in Figure 1. It differs from the original method in lines 6 and 7, where pre-pivoting and regular QR decomposition replace QRCP to ensure that for the permuted matrix $\tilde{C}$, we have $\|\tilde{C}_{1:n,k}\|_2 \geq \|\tilde{C}_{1:n,l}\|_2$ for all $1 \leq k < l \leq n$.

Matrix clustering is often used to reduce additional computational costs associated with involving pivoted versions of QR decomposition in the stratification algorithms. Using traditional matrix multiplication methods, the set of input matrices $\tilde{B}_i$ is obtained by multiplying p consecutive matrices for each i:

$$\tilde{B}_i = B_{ip} B_{ip-1} \dots B_{(i-1)p+1}, \quad 1 \leq i \leq \frac{L}{p} = \tilde{L} \tag{18}$$

These products are then used in the calculation of G, reducing the number of QR factorizations in the algorithm Figure 1 by a factor p. Typically, $p \sim 10$ is able to produce a significant performance boost while maintaining acceptable numerical properties.

3.2. *Time-dependent Green's function*

While in principle one can calculate the time-dependent Green's function by inverting $\mathcal{M}$ in Equation (17), the process is computationally extremely expensive if done naively. Moreover, in large-scale DQMC simulations, the number of time slices is typically $L \sim 10^2$

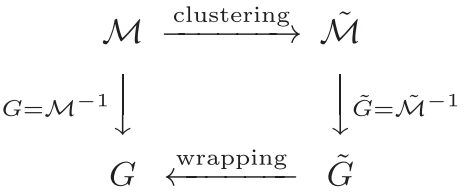


Figure 2. The computation of time-dependent Green's function $G = \mathcal{M}^{-1}$ can be divided into three steps: clustering, inversion, and wrapping.

and the number of sites is $N \sim 10^3$. Thus, matrices $\mathcal{M}$ have the dimension of order $\sim 10^5$, and require ~ 75 GB of memory for double precision real numbers. In order to maintain high performance for inversion of these relatively big matrices repeated hundreds of times in the simulation loop, the Hubbard matrices $\mathcal{M}$ are structure-preserving reduced before inversion into the matrices

$$\tilde{\mathcal{M}} = \begin{pmatrix} I & 0 & 0 & \cdots & 0 & \tilde{B}_1 \\ -\tilde{B}_2 & I & 0 & \cdots & 0 & 0 \\ 0 & -\tilde{B}_3 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\tilde{B}_{\tilde{L}} & I \end{pmatrix}.$$

where $\tilde{B}_i$ are defined as in Equation (18). This reduction is called clustering and the resulting matrix has dimension $(NL/p) \times (NL/p)$. Then, the matrix $\tilde{\mathcal{M}}$ is inverted, generating a subset of time-dependent Green's functions. The missing Green's function can be accessed via a process called wrapping. Such 'clustering-inversion-wrapping' work flow is depicted in Figure 2.

It should be pointed out that, even for the reduced matrix $\tilde{\mathcal{M}}$, direct inversion is still computationally expensive. In the conventional inversion methods, $\tilde{\mathcal{M}}$ is typically treated as a dense matrix and inverted via pivoted LU decomposition. Although this method guarantees a stable solution, it does not exploit the sparsity of $\tilde{\mathcal{M}}$, and thus requires redundant computational work. On the other hand, fast structure-exploiting inversion algorithms based on Gaussian elimination without or with partial pivoting, like cyclic reduction, are known to be unstable [30]. A blocked structured orthogonal factorization (BSOF) offers a stable and computationally efficient alternative to Gaussian elimination based inversion approaches [31]. BSOF-based inversion includes three successive phases – blocked structured orthogonal factorization, inversion of the block upper triangular factor R and the application of orthogonal factors to the inverse of R. We refer to this novel inversion method as blocked structured orthogonal inversion (BSOI).

In the factorization phase, we use the 'serial' version of the structured orthogonal factorization algorithm introduced in [30], which is proved to be stable and has an identical serial complexity to the best known blocked structure-exploiting factorization algorithms based on Gaussian elimination. The essence of this algorithm is in transformation of the matrix $\tilde{\mathcal{M}}$ through $\tilde{L} - 1$ block row updates (see algorithm Figure 3). It results in a block structured orthogonal factorization of the original matrix

$$\tilde{\mathcal{M}} = QR, \tag{19}$$

Downloaded by [University of California Davis] at 09:40 27 April 2016

procedure CLUSTTIMEDEPENDENTG($\{\tilde{B}_i | 1 \le i \le \tilde{L}\}$)
▷ **Phase I**. Blocked structured orthogonal factorization
  $R \leftarrow O$; $\tilde{R}_{1,1} \leftarrow I$; $\tilde{R}_{1,\tilde{L}} \leftarrow \tilde{B}_1$
  for $k \in \{1, 2, ..., \tilde{L} - 2\}$ **do**
    Compute QR: $Q^{(k)} \begin{pmatrix} R_{k,k} \\ O \end{pmatrix} = \begin{pmatrix} \tilde{R}_{k,k} \\ -\tilde{B}_{k+1} \end{pmatrix}$
    $\begin{pmatrix} R_{k,k+1} & R_{k,\tilde{L}} \\ \tilde{R}_{k+1,k+1} & \tilde{R}_{k+1,\tilde{L}} \end{pmatrix} \leftarrow (Q^{(k)})^T \begin{pmatrix} O & \tilde{R}_{k,\tilde{L}} \\ I & O \end{pmatrix}$
  end for
  Compute QR:
  $Q^{(\tilde{L}-1)} \begin{pmatrix} R_{\tilde{L}-1,\tilde{L}-1} & R_{\tilde{L}-1,\tilde{L}} \\ O & R_{\tilde{L},\tilde{L}} \end{pmatrix} = \begin{pmatrix} \tilde{R}_{\tilde{L}-1,\tilde{L}-1} & \tilde{R}_{\tilde{L}-1,\tilde{L}} \\ -\tilde{B}_{\tilde{L}} & I \end{pmatrix}$
▷ **Phase II**. Blocked back substitution (row version)
  $\tilde{G} \leftarrow O$; $\tilde{G}_{\tilde{L}-2:\tilde{L},\tilde{L}-2:\tilde{L}} \leftarrow R^{-1}_{\tilde{L}-2:\tilde{L},\tilde{L}-2:\tilde{L}}$
  Batched$_{i=1:\tilde{L}-3}$ $\{\tilde{G}_{ii} \leftarrow R^{-1}_{ii}\}$
  $\tilde{G}_{1:\tilde{L}-3,\tilde{L}} \leftarrow R_{1:\tilde{L}-3,\tilde{L}} \tilde{G}_{\tilde{L},\tilde{L}}$
  Batched$_{i=1:\tilde{L}-3}$ $\{\tilde{G}_{i,i+1} \leftarrow -\tilde{G}_{ii} R_{i,i+1}\}$
  Batched$_{i=1:\tilde{L}-3}$ $\{\tilde{G}_{i,\tilde{L}} \leftarrow -\tilde{G}_{ii} R_{i,\tilde{L}}\}$
  for $i \in \{\tilde{L} - 3, \tilde{L} - 4, ..., 1\}$ **do**
    $\tilde{G}_{i,i+2:\tilde{L}} \leftarrow \tilde{G}_{i,i+2:\tilde{L}} + \tilde{G}_{i,i+1} \tilde{G}_{i+1,i+2:\tilde{L}}$
    $\tilde{G}_{i,i+1} \leftarrow \tilde{G}_{i,i+1} \tilde{G}_{i+1,i+1}$
  end for
    ▷ **Phase III**. Applying Householder reflectors
  for $k \in \{\tilde{L} - 1, \tilde{L} - 2, ..., 1\}$ **do**
    $\tilde{G}_{1:\tilde{L},k:k+1} \leftarrow \tilde{G}_{1:\tilde{L},k:k+1} (Q^{(k)})^T$
  end for
  return $\tilde{G}$
end procedure

Figure 3. Blocked structured orthogonal inversion of clustered Hubbard matrix $\tilde{\mathcal{M}}$.

where Q is a product of the orthogonal matrices $Q^{(k)}$ extended by identity blocks

$$Q = \prod_{k=1}^{\tilde{L}-1} Q_k = \prod_{k=1}^{\tilde{L}-1} I_{N(k-1)} \oplus Q^{(k)} \oplus I_{N(\tilde{L}-k-1)}, \tag{20}$$

and R has a block upper bidiagonal form with a dense final block column

$$R = \begin{pmatrix} R_{11} & R_{12} & 0 & \cdots & R_{1,\tilde{L}} \\ 0 & R_{22} & R_{23} & \cdots & R_{2,\tilde{L}} \\ 0 & 0 & R_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & R_{\tilde{L}-1,\tilde{L}} \\ 0 & 0 & 0 & \cdots & R_{\tilde{L},\tilde{L}} \end{pmatrix}. \tag{21}$$

Thus, the Green's function can be obtained by

$$\tilde{G} = \tilde{\mathcal{M}}^{-1} = R^{-1} \prod_{k=\tilde{L}-1}^{1} Q_k^T. \tag{22}$$

Downloaded by [University of California Davis] at 09:40 27 April 2016

R^{-1} is computed via blocked back substitution (BBS) algorithms [32] using the block structure pattern Equation (21) of matrix R. The column version of inversion via BBS requires $\sim 2\tilde{L}^2N^3$ flops, whereas the computational cost of the row version is roughly a factor of two less, $\sim \tilde{L}^2N^3$ flops. On the other hand, the column-wise inversion algorithm requires only block rows from 1 to $k-1$ in order to compute the kth block column, thus providing us with an opportunity to perform factorization of $\mathcal{M}$ and inversion of R in parallel. In contrast, merging factorization and inversion phases is impossible if the row-wise algorithm is used. Due to the specific structure of orthogonal matrices Q_k, computing the product in Equation (22) is equivalent to successive applying Householder reflectors to the block column pairs of R^{-1} from the right in a backward order.

Before presenting benchmark results, we want to comment that the optimal choice of p when forming $\mathcal{M}$ is a balance between a number of conflicting issues. For example, using large p values (corresponding to small and dense matrices) can reduce the cost of computing the determinant, which scales as the cube of the matrix size. However, to take advantage of modern hybrid multicore CPU and GPU platform, it is more desirable to spread out the computation of large matrices over different computational cores. From this point of view, a small p seems to be a preferable choice.

3.3. *Performance on multicore processors and GPUs*

In this subsection, we present benchmark results of the algorithms described in Secs. III-A and B for the calculation of equal-time and time-dependent Green's functions on multicore processors and GPUs. Computations reported in this paper are carried out on two sockets of Intel Xeon X5670 with Westmare-EP micro-architecture coupled with Nvidia GeForce GTX 480 graphic card. Intel MKL, MAGMA and Nvidia CUBLAS libraries are used for BLAS and LAPACK routines in our implementations. All computations are in double precision. The peak performance of BLAS matrix-matrix multiplication (MMM) routine `DGEMM` observed on the target platform is 142 GFlops for MKL and 167 GFlops for CUBLAS.

To verify the advantages of the stratification method with pre-pivoting for computing the equal-time Green's function G, we compare its performance with the performance of the classic Loh algorithm based on QRCP (Figure 4). The blue curve in Figure 4 represents peak performance of MKL BLAS routine `DGEMM` for the multiplication of two square $n^D \times n^D$ matrices. It indicates the practical upper bound of performance of any algorithm used to calculate equal-time Green's function on the given architecture.

This comparison demonstrates that using pre-pivoting dramatically improves performance of the stratification algorithms on multicore platforms, allowing up to 70% increase in the `DGEMM` performance rate. Moreover, experimental studies show that stratification methods with pre-pivoting and with column pivoting produce results with similar accuracy [29]. Performance results reported in Ref. [29] indicate the great potential of the pre-pivoted stratification algorithms for execution on hybrid CPU+GPU platforms. Since most of the computational work in equal-time DQMC simulations is related to computing Green's function, use of pre-pivoting significantly reduces total simulation time. Figure 5 illustrates decrease of total simulation time if QR with pre-pivoting replaces QRCP in Loh's algorithm.

In order to examine the BSOI algorithm for computing the time-dependent Green's functions, we developed implementations for multicore CPU, GPU and hybrid CPU+GPU

Downloaded by [University of California Davis] at 09:40 27 April 2016

Downloaded by [University of California Davis] at 09:40 27 April 2016

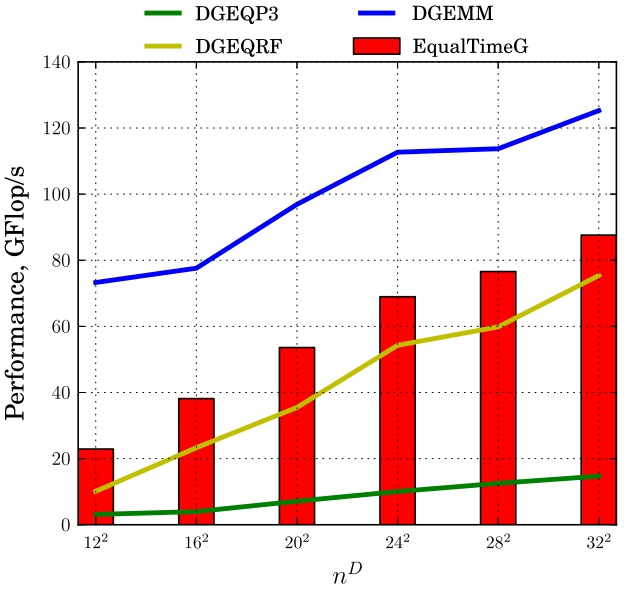


Figure 4. (Colour online) Performance of equal-time Green's function computation.

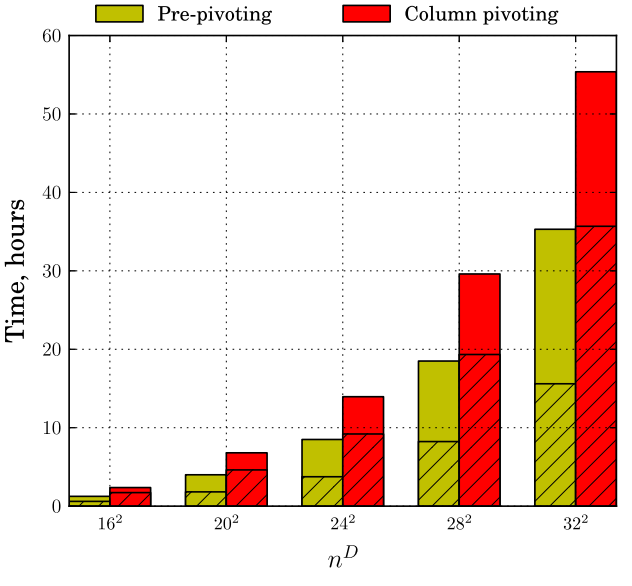


Figure 5. (Colour online) Time for a complete equal-time DQMC simulation. Dashed portions of the bars correspond to the time spent on computing Green's function.

processing, respectively. The BSOI solvers receive the matrix $\tilde{\mathcal{M}}$ in an unpacked form as input and perform in-place inversion. All three implementations are based on the use of subroutines from BLAS and LAPACK interfaces. Our CPU and GPU codes implement the

phases of BSOI separately in three subroutines called successively. In both cases, the row version of the BBS algorithm is used for inversion of the factor R. In contrast, in our hybrid CPU+GPU implementation, we merge the phase of orthogonal factorization with blocked back substitution into a single subroutine and use the combination of row and column versions in BBS phase. This allows us to exploit the GPU for matrix-matrix multiplications during BBS in parallel with the CPU employed in factorization, as is done in Ref. [31].

In Figures 6 and 7, we compare the BSOI approach to inversion algorithms based on LU factorization with complete pivoting (LUPI) and conventional QR (QRI) factorization without pivoting where the reduced Hubbard matrix $\tilde{\mathcal{M}}$ is treated as dense. These figures also present the timing breakdown between the different inversion phases. The bottom portions of the bars correspond to the factorization phase, whereas the hatched upper parts represent the time fractions for the further inversion steps. Both figures demonstrate significant speed-up of BSOI with respect to methods based on dense $\tilde{\mathcal{M}}$, if the value L/p increases. The speed-up grows with the number of sites $N = n^D$. The major contribution in improvement of BSOI with respect to QR-based inversion is related to the calculation of R^{-1}.

Figure 8 illustrates the performance of BSOI codes on different platforms. Hybrid CPU+GPU implementation is up to 1.7× faster over the best of CPU and GPU codes. Its peak performance is higher than the practical peak performance of `DGEMM` on CPU and is only 1.1× lower than the peak of `DGEMM` on GPU. Moreover, the difference in performance in the interval $64 \leq N \leq 1024$ does not exceed 1.5× for our hybrid CPU+GPU implementation. We conclude that this code works equally efficiently for Hubbard matrices $\tilde{\mathcal{M}}$ with a diverse range of the number of sites N.

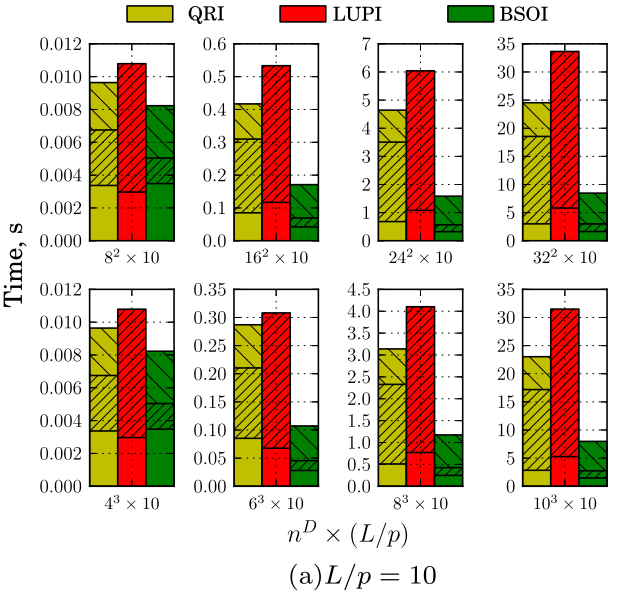
Figure 9 shows that the accuracy of BSOI is comparable with LU-based inversion, and lies on the double precision resolution boundary.

We conclude this section by summarizing the overall status of our QMC software. As shown in Figure 5, pre-pivoting provides a substantial speed-up on a complete DQMC simulation which is restricted to the evaluation of equal-time correlation functions. What our latest work employing the BSOI technique shows (Figures 6 and 7) is that the inversion step of $\tilde{M}$ can similarly be made much more efficient. However, experiments demonstrate that in DQMC simulations with time-dependent measurements, the majority of computational time is spent on the wrapping step (see Figure 2). Figure 10 shows the total simulation time for different number of sites ($L = 160$, $p = 10$) if BSOI and conventional LUPI are used to calculate clustered time-dependent Green's functions. Therefore, in order to boost overall performance of the DQMC simulation involving the time-dependent measurements, the wrapping algorithm will need to be improved. This will be an important topic for future developments.

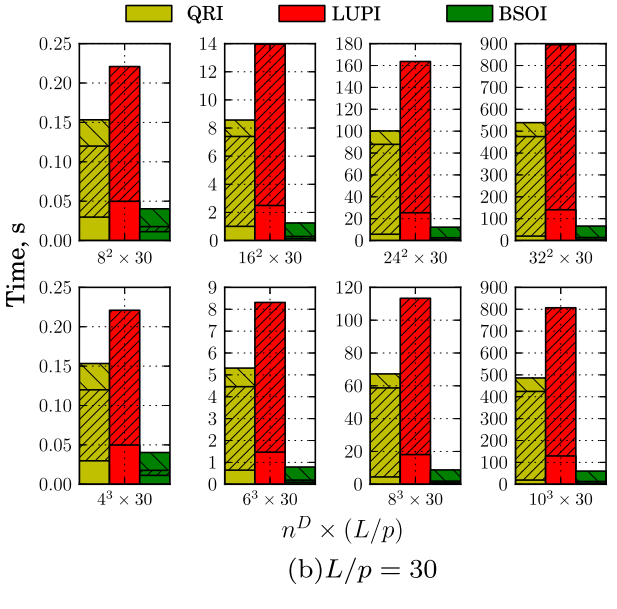
4. Selected applications

We conclude this paper by exhibiting a few of the applications of DQMC simulations of the Hubbard Hamiltonian. We focus on inhomogeneous systems for which the Hamiltonian creates spatial regions with differing magnetic, superconducting and charge fluctuation properties. For such situations the challenge of achieving large enough lattice sizes is especially acute, as opposed to homogeneous ones where translation invariance is present. As noted at the end of the preceding section, improving the efficiency of DQMC involves both speeding up those parts of the approach which involve equal time correlations only and, separately, those that require unequal-time measurements (such as

Downloaded by [University of California Davis] at 09:40 27 April 2016



(a) $L/p = 10$



(b) $L/p = 30$

Figure 6. (Colour online) Inversion time of $\tilde{\mathcal{M}}$ for 2D and 3D lattices with fixed L/p. Notation: QRI – inversion based on QR factorization without pivoting, LUPI – inversion based on LU factorization with complete pivoting, BSOI – block structured orthogonal inversion. The bottom portions of the bars correspond to the factorization phase, the hatched upper parts correspond to the further inversion steps.

spectral functions and susceptibilities). The results presented here include our improvements to the equal-time algorithm, but not to the unequal-time one.

Downloaded by [University of California Davis] at 09:40 27 April 2016

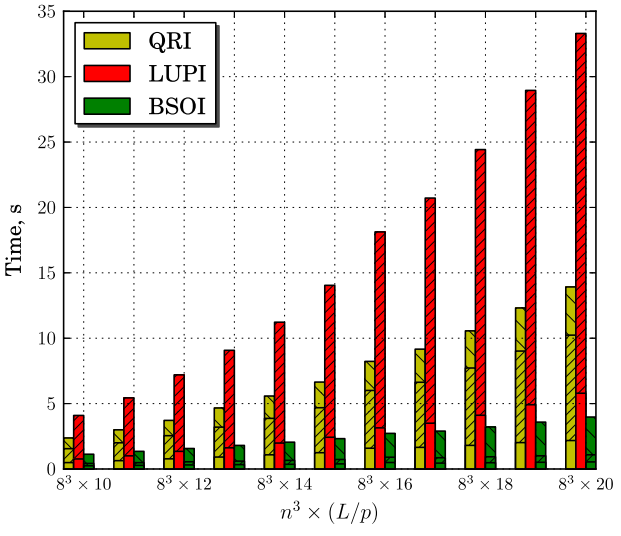


Figure 7. (Colour online) Inversion time of $\tilde{\mathcal{M}}$ for $8 \times 8 \times 8$ lattice with different L/p.

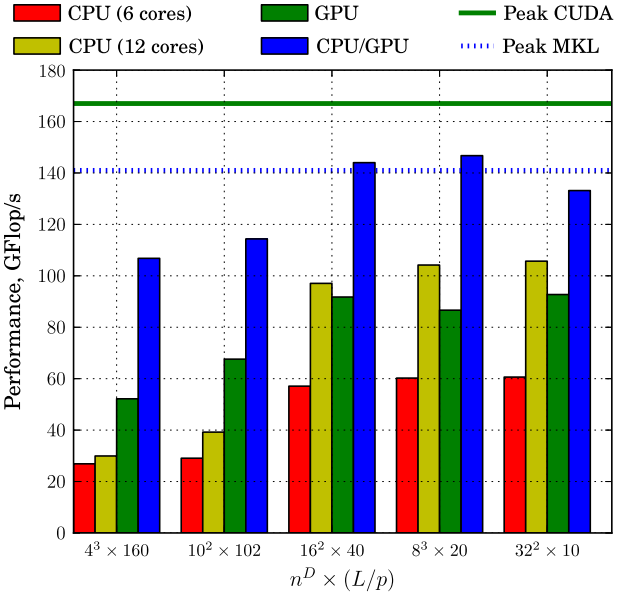


Figure 8. (Colour online) Performance of BSOI implementations on different platforms ($n^D \times L/p = 10240$).

In condensed matter systems, insulating behaviour can arise in a variety of ways. The most simple mechanism is a 'band insulator' (BI) in which there is a gap in the allowed energy levels and the particle number is such that the energy levels below the gap are filled and those above the gap are empty. When, instead, the Fermi level is in the middle of a

Downloaded by [University of California Davis] at 09:40 27 April 2016

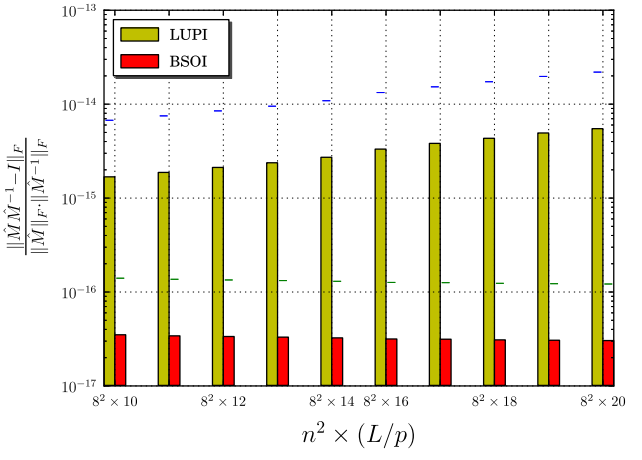


Figure 9. (Colour online) Relative error of inversion for an 8×8 lattice. The error is estimated via Frobenius norm.

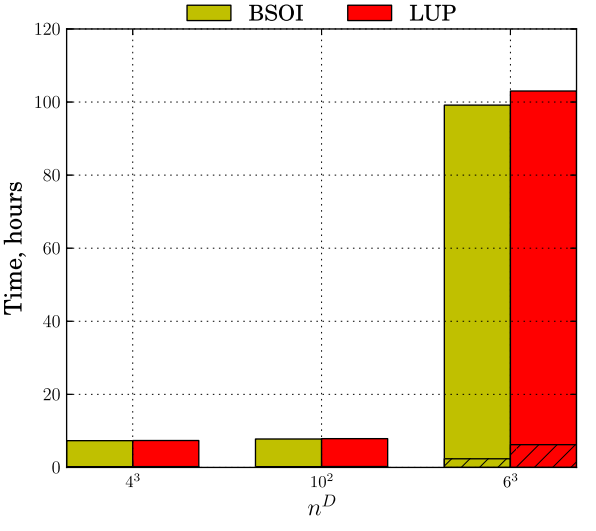


Figure 10. (Colour online) Time for a complete time-dependent DQMC simulation if $L = 160$, $p = 10$. Dashed portions of the bars correspond to the time spent on computing clustered Green's function $\tilde{G}$.

band, one can still get an insulating phase if the interactions are strong enough that they restrict particle motion, a 'Mott Insulator' (MI), or if randomness localizes the particles, an 'Anderson insulator' (AI). Often the MI is accompanied by other types of order, e.g. long range magnetic correlations.

We have recently studied a model which has the possibility of both BI and MI behaviour. It consists of two layers whose individual energy bands are inverted relative to one another, so that the momenta at which one layer has a band maximum is at a band minimum for

Downloaded by [University of California Davis] at 09:40 27 April 2016

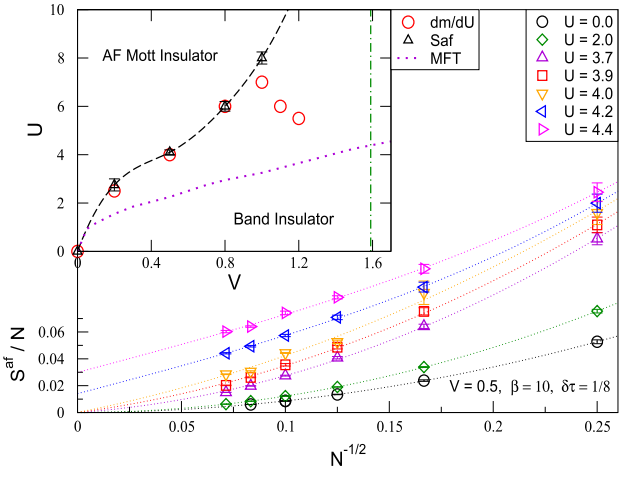


Figure 11. (Colour online) Inset: Phase diagram of a correlated band insulator. V is the interlayer hopping and U is the on site interaction. Cross-overs from the MI and BI are shown as dashed and dotted lines. The open circles mark the places where the derivative of the density with respect to on-site repulsion dn/dU is maximal. Main Figure: finite size scaling results for magnetic correlations at different U and fixed V.

the other and vice-versa. When electrons are allowed to hop between these layers, via a hybridization V, the result is a band insulator. The corresponding Hamiltonian would look much like Equation (1), except the hopping t is not the same between all near neighbour sites $\langle ij \rangle$. Rather it has values t in one layer, $-t$ in the other and V between the layers.

The resulting phase diagram [33] is shown in the inset to Figure 11. In the region marked as AF Mott Insulator, the system is insulating and also has long range antiferromagnetic (AF) correlations, as determined by the extrapolation to the thermodynamic limit $1/\sqrt{N} \to 0$ in the main figure. At fixed $V = 0.5$ there are long range AF correlations for $U > 4$. Studies of the spectral function reveal the presence of a gap throughout the phase diagram. The competition between BI and MI does not produce metallic behaviour anywhere, unlike other models [34]. Notice that achieving large lattice sizes is crucial to obtaining reliable extrapolation of the physical properties in the thermodynamic limit.

A second application of our new DQMC implementation is to ultracold fermionic atoms confined in a trap. In these systems, interfering laser beams generate an 'optical lattice' analogous to that provided by the crystalline arrangement of atomic nuclei in a conventional solid. To prevent the atoms from escaping, however, a quadratic potential $\mu_i = \mu_0 - V_{\text{trap}}\, i^2$ is applied, which makes it energetically unfavourable to move away from the centre of the lattice. As a consequence, the density, and charge and pair correlations, vary smoothly through the lattice. The conventional way to study such situations is through the 'local density approximation' (LDA) in which the properties of the system at a particular spatial location are assumed to be those of a homogeneous system of the same (uniform) density.

Pairing correlations are shown in Figure 12. The vertical line at $|i| \approx 9$ shows the spatial location where the density is one fermion per site. For uniform systems, and hence within the LDA (lines), superconductivity is suppressed at this density, owing to the strong competition of charge order. Our lattices are now large enough that we can directly simulate the inhomogeneous system (symbols), which show no such suppression. This is one of the

Downloaded by [University of California Davis] at 09:40 27 April 2016

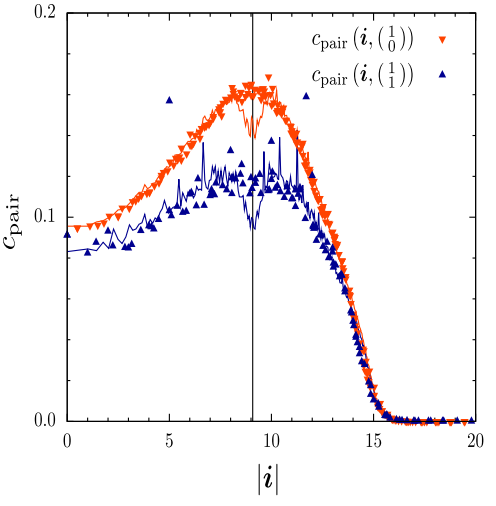


Figure 12. (Colour online) Pairing correlations in a confined fermionic system of 564 particles interacting with an attractive $U = -6$ with $\mu_0 = -0.8$, $V_{\rm trap} = 0.0097$ and $\beta = 9$. The symbols are the result of simulations of the full inhomogeneous system, while the lines are the result of a calculation within the local density approximation. The latter approach is seen to fail to describe the inhomogeneity accurately in the region around $i = 9$ where the density $\rho = 1$.

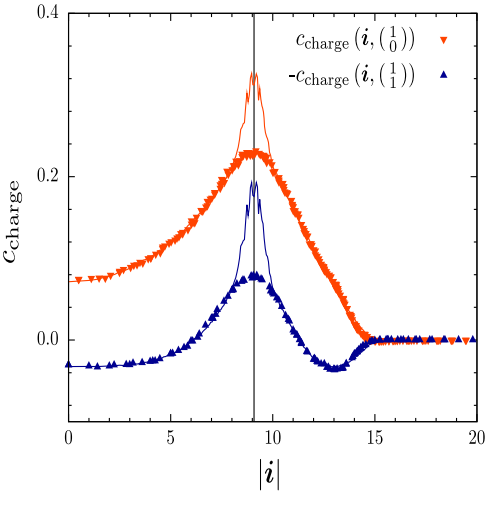


Figure 13. (Colour online) Same parameters as for Figure 12. Here the charge correlations are shown. Again, the LDA is seen to fail, vastly overestimating the degree of charge order in the region where $\rho = 1$.

first, and most dramatic, demonstrations of the failure of the LDA in modelling such systems. Figure 13 provides a counterpart to Figure 12, exhibiting the charge correlations. The sharp LDA peaks (lines) are absent in the full simulation (symbols).

Downloaded by [University of California Davis] at 09:40 27 April 2016

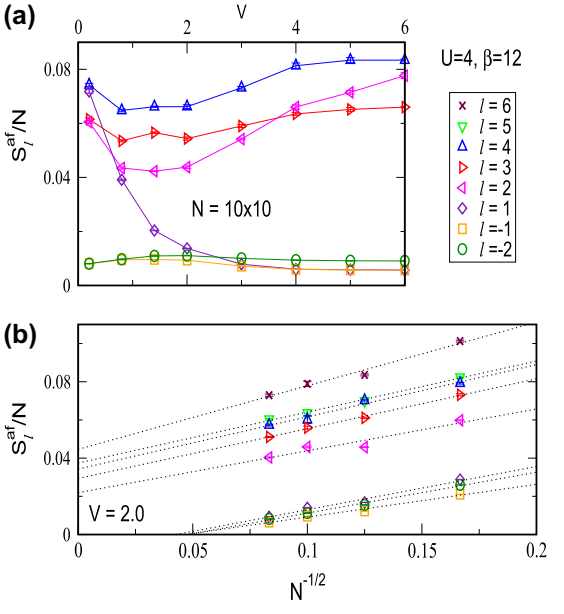


Figure 14. (Colour online) Top panel: Antiferromagnetic structure factor $S_l^{\rm af}$ for two layers $l = -2, -1$ with $U = 0$ in contact via a hopping V with six layers $l = 1, 2, \cdots, 6$ with $U = 4$. As V increases, $S_l^{\rm af}$ is reduced for all positive l, but it then recovers except for $l = 1$, the layer most immediately adjacent to the $U = 0$ metal. The bottom panel provides data for different lattice sizes at $V = 2$.

In Figure 11, we considered a model with two layers (or, alternately viewed, bands). Our final application concerns a many layer system. Here, the interest is in studying the interface between different materials, e.g. one magnetic and one non-magnetic, or one metallic and the other insulating. Such interfaces were impossible to study when lattices were limited in size to $N \sim 10^2$. Now, however, we are able to study eight 10×10 layers and interfaces between them. Figure 14 shows a specific example. Two layers $l = -2, -1$ have weak interactions ($U = 0$) between the electrons. As a consequence, their magnetic structure factors $S_l^{\rm af}$ are small. Above these non-magnetic layers we place six more strongly interacting ($U = 4$) layers $l = 1, 2, \cdots, 6$. We then vary the ability of electrons to tunnel between layers $l = -1$ and $l = 1$ which connect the materials, i.e. we vary the hybridization V which is the t_{ij} for adjacent sites of these two layers. Figure 14 illustrates what happens as V increases. Contact with the non-magnetic metallic layers $l = -2, -1$ at first decreases magnetism in layers $l = 1, 2, \cdots, 6$. Eventually, however, only layer $l = 1$ completes the transition to paramagnetism. Magnetism in the other interacting layers $l = 2, 3, \cdots, 6$ recovers.

The physical interpretation of this non-monotonicity is that large hopping V ultimately results in the formation of singlets across the $l = -1, 1$ interface. These singlets then screen layers $l = 2, 3, \ldots, 6$ from the metal, leading to a recovery of the magnetism. A finite size scaling analysis (bottom panel) indicates that when the recovery commences at $V \approx 2$ there is long range order in layers $l = 2, 3, \cdots, 6$, but not layers $l = -2, -1, 1$.

These three examples illustrate the power of the new DQMC algorithms being developed. The key shared feature is that they involve lattices with different spatial

Downloaded by [University of California Davis] at 09:40 27 April 2016

regions in contact: two bands of opposite curvature, spatially varying density, or weakly and strongly correlated materials. Large N is essential to modeling such situations.

5. Conclusions

In this paper, we have described recent algorithmic advances in Determinant Quantum Monte Carlo whose goal is to formulate efficient and robust linear algebra kernels which can take advantage of multicore and GPU architectures. We have demonstrated an order of magnitude speed-up of our codes, without compromising the numerical stability, and have benchmarked the performance for different spatial lattice sizes and inverse temperatures. The end result is, roughly speaking, an order of magnitude increase in performance on a typical 12 core + Nvidia Fermi GPU machine.

An important development over the past several years, which has also been used in DQMC and related algorithms, is the construction of 'delayed' [35] and 'submatrix' [36,37] update methods. Roughly speaking, these approaches take advantage of the fact that only a partial knowledge of the new M^{-1} which arises from the updating of a given Hubbard-Stratonovich variable is needed in updating the next one. Thus, cpu time can be saved by delaying the calculation of unneeded elements of the new M^{-1} until enough of them have accumulated to make doing so more efficient. A future step in the development of our codes will be to coordinate the use of delayed and submatrix updating with the methods described in this manuscript.

We have also provided some recent results of DQMC simulations. Most of these, in fact, have been obtained with the previous version of our code, which made use of a variety of novel algorithmic upgrades, did not yet incorporate the changes which have been described here, and were run on simple single core architectures without GPUs. We estimate an additional order of magnitude or more speed-up is now possible, and will enable lattice sizes with at least twice as many sites to be studied.

Downloaded by [University of California Davis] at 09:40 27 April 2016

Acknowledgements

This work was supported by the National Science Foundation under grant NSF-PHY-1005503. S. Chiesa, A. Euverte, G.G. Batrouni, and E. Assmann played important roles in testing the codes by applying them to different Hamiltonians and geometries. We also thank B. Gibbs for useful input. S. Gogolenko would also like to thank the Fulbright Program Office in Ukraine and the Institute of International Education for finantial support during this study.

Notes

1. Since the chemical potential term is diagonal, it is often combined with the last term of Equation (8). Therefore $\mathcal{T}_\sigma$ contains the hopping matrix elements only.
2. When the spin degree of freedom is included in the formulation, the occupation number basis consists of direct product of basis state at different spin sectors. Consequently, the trace becomes a product of traces of each spin component.

References

[1] R. Blankenbecler, R.L. Sugar and D.J. Scalapino, Phys. Rev. D24 (1981) p.2278.
[2] T. Valla, A. Fedorov, J. Lee, J. Davis and G. Gu, Science 314 (2006) p.1914.

[3] J.M. Tranquada, G.D. Gu, M. Hucker, Q. Jie, H.-J. Kang, R. Klingeler, Q. Li, N. Tristan, J.S. Wen, G.Y. Xu, et al., Phys. Rev. B78 (2008) p.174529.
[4] E. Arrigoni, M. Zacher, R. Eder, W. Hanke, A. Harju and S. Kivelson, J. Phys. Chem. Solids 63 (2002) p.2207.
[5] A. Cho, Science 320 (2008) p.312.
[6] D. Vollhardt, in *Correlated Electron Systems*, V. J. Emery, ed., Vol. 57, World Scientific, Singapore, 1993.
[7] A. Georges, G. Kotliar, W. Krauth and M. Rozenberg, Rev. Mod. Phys. 68 (1996) p.13.
[8] M.H. Hettler, A.N. Tahvildar-Zadeh, M. Jarrell, T. Pruschke and H.R. Krishnamurthy, Phys. Rev. B58 (1998) p.R7475.
[9] Th. Maier, M. Jarrell, T. Pruschke and M.H. Hettler, Rev. Mod. Phys. 77 (2005) p.1027.
[10] E. Kozik, E. Burovski, V.W. Scarola and M. Troyer, Phys. Rev. B 87 (2013) p.205102.
[11] G. Sugiyama and S.E. Koonin, Ann. Phys. 168 (1986) p.1.
[12] S. Sorella, S. Baroni, R. Car and M. Parrinello, Euro. phys. Lett. 8 (1989) p.663; S. Sorella, E. Tosatti, S. Baroni, R. Car, and M. Parrinello, Int. J. Mod. Phys. B1 (1989) p.993.
[13] S.R. White, D.J. Scalapino, R.L. Sugar, E.Y. Loh, J.E. Gubernatis and R.T. Scalettar, Phys. Rev. B40 (1989) p.506.
[14] J.E. Gubernatis, E.Y. Loh, D.J. Scalapino, R.L. Sugar, R.T. Scalettar, and S.R. White, in Proceedings of the Los Alamos Conference on Quantum Simulation, Aug. 8–11, 1989, World Scientific, Los Alamos, NM, 1990.
[15] E.Y. Loh, J.E. Gubernatis, R.T. Scalettar, S.R. White, D.J. Scalapino and R.L. Sugar, Phys. Rev. B41 (1990) p.9301.
[16] E.Y. Loh, J.E. Gubernatis, R.T. Scalettar, S.R. White, D.J. Scalapino and R.L. Sugar, Intl. J. Mod. Phys. 16 (2005) p.1319.
[17] M. Troyer and U.-J. Wiese, Phys. Rev. Lett. 94 (2005) p.170201.
[18] Z. Bai, W. Chen, R. Scalettar and I. Yamazaki, *Numerical methods for quantum Monte Carlo simulations of the Hubbard model*, in *Multi-Scale Phenomena in Complex Fluids*, T. Y. Hou, C. Liu and J.-G. Liu, eds., Higher Education Press, Beijing, 2009, p.1.
[19] H.F. Trotter, Proc. Amer. Math. Soc. 10 (1959) p.545.
[20] M. Suzuki, Prog. Theor. Phys. 56 (1976) p.1454.
[21] R.M. Fye, Phys. Rev. B33 (1986) p.6271.
[22] As a practical matter, choosing $\epsilon \sim 0.1$ will typically yield results for many measurements whose 'Trotter error' is only a few percent, often no larger than the statistical error. In some cases, however, when higher accuracy is needed, the extrapolation $\epsilon \to 0$ must be done. See, for example E.V. Gorelik, D. Rost, T. Paiva, R. Scalettar, A. Klümper and N. Blümer, Phys. Rev. A85 (2012) p.061602.
[23] J.E. Hirsch, Phys. Rev. B31 (1985) p.4403.
[24] G.G. Batrouni and R.T. Scalettar, Phys. Rev. B42 (1990) p.2282.
[25] L. Chen and A.-M.S. Tremblay, Int. J. Mod. Phys. B6 (1992) p.547.
[26] M. Hohenadler, Z.Y. Meng, T.C. Lang, S. Wessel, A. Muramatsu and F.F. Assaad, Phys. Rev. B85 (2012) p.115132.
[27] T. Paiva, Yen Lee Loh, N. Trivedi, M. Randeria and R.T. Scalettar, Phys. Rev. Lett. 107 (2011) p.086401.
[28] E.Y. Loh and J.E. Gubernatis, *Stable Numerical Simulations of Models of Interacting Electrons in Condensed Matter Physics*, in *Electronic Phase Transitions*, Modern Problems in Condensed Matter Sciences, W. Hanke and Yu.V. Kopaev, eds., Ch. 4, Elsevier Science Publishers, Amsterdam, 1992, p.177.
[29] A. Tomas, C.-C. Chang, R. Scalettar and Z. Bai, in Proceedings of 26th IEEE International Parallel and Distributed Processing, Symposium, Shanghai, 2012, p.308.
[30] S.J. Wright, SIAM J. Sci. Stat. Comput. 13(3) (1992) p.742.

Downloaded by [University of California Davis] at 09:40 27 April 2016

[31] S. Gogolenko and Z. Bai, Tech. Rep. CSE-2013-78, Department of Computer Science, University of California, Davis, CA, 2013.
[32] G.H. Golub and C.F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimire, 1996.
[33] A. Euverte, S. Chiesa, R.T. Scalettar and G.G. Batrouni, Phys. Rev. B 87 (2013) p.125141.
[34] K. Bouadim, N. Paris, F. Hebert, G.G. Batrouni and R.T. Scalettar, Phys. Rev. B76 (2007) p.085112.
[35] G. Alvarez, M.S. Summers, D.E. Maxwell, M. Eisenbach, J.S. Meredith, J.M. Larkin, J. Levesque, T.A. Maier, P.R.C. Kent, E.F. D'Azevedo and T.C. Schulthess, in SC 08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, IEEE Press Piscataway, NJ, USA, 2008, p. 110.
[36] P.K.V.V. Nukala, T.A. Maier, M.S. Summers, G. Alvarez and T.C. Schulthess, Phys. Rev. B80 (2009) p.195111.
[37] E. Gull, P. Staar, S. Fuchs, P. Hukala, M.S. Summers, T. Pruschke, T.C. Schulthess and T. Maier, Phys. Rev. B83 (2011) p.075122.

Downloaded by [University of California Davis] at 09:40 27 April 2016