Multi-Length Scale Matrix Computations

and Applications in Quantum Mechanical Simulations

Zhaojun Bai
http://www.cs.ucdavis.edu/~bai

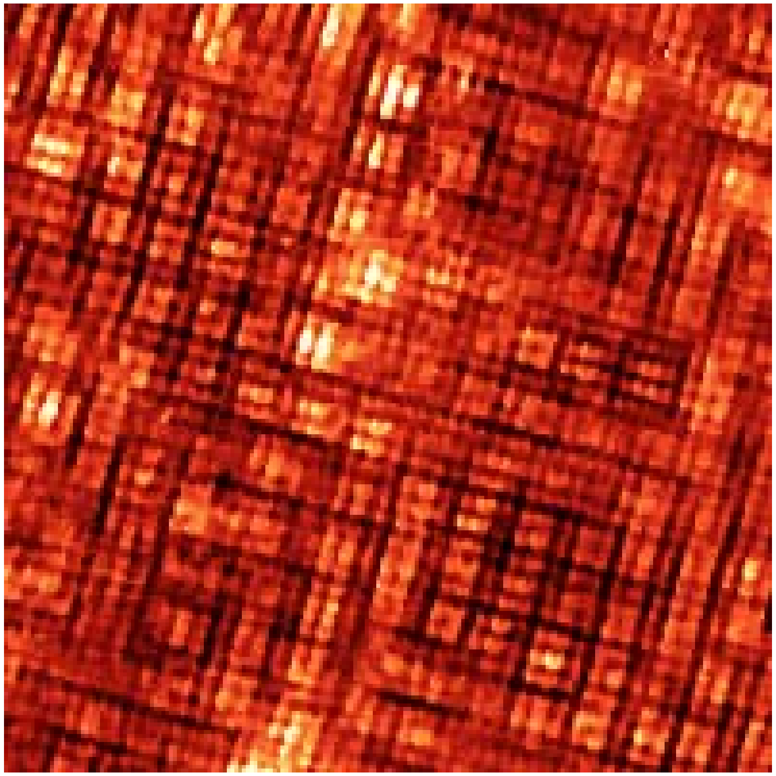
joint work with

Wenbin Chen, Roger Lee, Richard Scalettar, Ichitaro Yamazaki

Workshop on Future Directions in Tensor-Based Computation and Modeling
NSF, Arlington, Feb. 20-21, 2009

Computational Material Science

Simulation and understanding properties of solid-state materials: magnetism, metal-insulator transition, high temperature superconductivity, ...



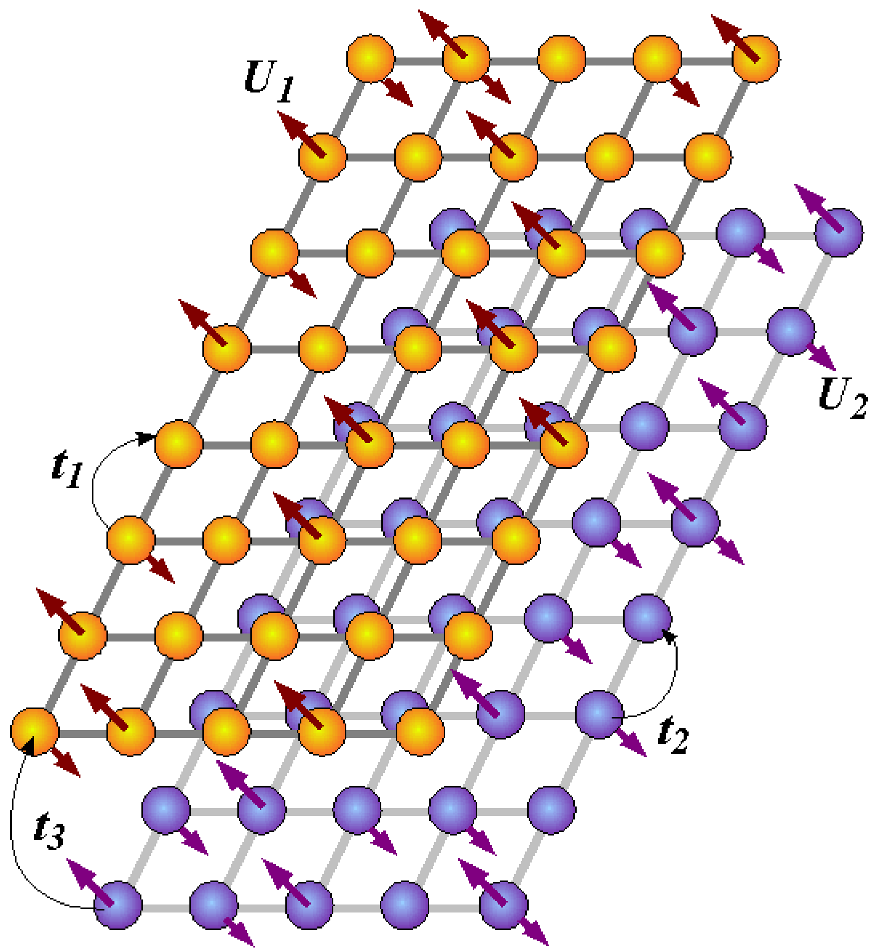
Conductance map of an electronic crystal state ... at the atomic scale. [T. Hanaguri *et al*, Nature 430, 1001 (2004)]

Outline of this talk

1. Hubbard model and quantum Monte Carlo simulation
2. QUEST: QUantum Electron Simulation Toolbox
3. Multi-length scale matrix computations – "tensor-based?"
   - Multi-length scale matrix analysis
   - Communication-avoiding stable matrix inversion
   - Self-adapting direct linear solvers
   - Robust preconditioned iterative solvers
4. Concluding remarks

Supported by NSF and DOE SciDAC

Many body simulation on multi-layer lattice



Hubbard model and quantum Monte Carlo simulation

Hubbard Hamiltonian – 4^N

$$
\begin{aligned}
\mathcal{H} &= \mathcal{H}_K + \mathcal{H}_\mu + \mathcal{H}_V \\
&= -t \sum_{\langle i,j\rangle,\sigma} (c^\dagger_{i\sigma} c_{j\sigma} + c^\dagger_{j\sigma} c_{i\sigma}) + \mu \sum_i (n_{i\uparrow} + n_{i\downarrow}) + U \sum_i (n_{i\uparrow} - \frac{1}{2})(n_{i\downarrow} - \frac{1}{2})
\end{aligned}
$$

- $\langle i, j\rangle$: a pair of nearest-neighbor spatial sites
- $\sigma = \uparrow, \downarrow$: spin direction of electrons
- $c^\dagger_{i\sigma}$ ($c_{i\sigma}$) creates (destroys) an electron of spin σ on site i
- t: hopping parameter $\Leftarrow$ kinetic energy
- U: local repulsion between electrons $\Leftarrow$ potential energy
- μ: controls the electron density $\Leftarrow$ chemical potential energy

Related quantum many body models:

- Ising model for phase transition
- Anderson model of localization for electron transport

Physical observable $\mathcal{E}$

The expected value of a physical observable $\mathcal{E}$, such as *density-density correlation*, *spin-spin correlation*, *the magnetic susceptibility*, is given by

$$\langle \mathcal{E} \rangle = \mathrm{Tr}\left(\mathcal{E} \mathcal{P} \right),$$

where $\mathcal{P}$ is the probability (Boltzmann) distribution

$$\mathcal{P} = \frac{1}{\mathcal{Z}} e^{-\beta \mathcal{H}}$$

and

$$\beta \propto \frac{1}{T} = \frac{1}{\text{Temperature}}$$

$$\mathcal{Z} = \mathrm{Tr}(e^{-\beta \mathcal{H}}) = \text{the partition function}$$

Computational approximations of Boltzmann distribution $\mathcal{P}$

$$\mathcal{P} = \frac{1}{\mathcal{Z}} e^{-\beta \mathcal{H}} \quad \xrightarrow{\text{"path integral"}} \quad \begin{cases} P(h) = \frac{1}{Z_h} \mathbf{det}[M_+(h)] \mathbf{det}[M_-(h)] \\ P(x, p, \Phi_\sigma) = \frac{1}{Z_H} e^{-H(x,p,\Phi_\sigma)} \end{cases}$$

[Feynman'65, ...]

- Determinant QMC

$$h \sim P(h)$$

- Hybird QMC (=molecular dynamics + mc)

$$(x; p, \Phi_\sigma) \sim P(x; p, \Phi_\sigma)$$

[Blankenbecler/Scalapino/Sugar'81, Hirsch'85, ...]

[Scalettar/Scalapino/Sugar/Toussaint'87, ...]

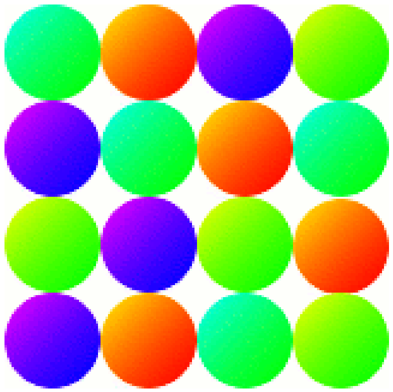
QUEST: QUantum Electron Simulation Toolbox

- Fortran 90 package for determinant (and hybrid) Monte Carlo simulations
- Integration and revision of several "legacy" codes developed in the past two decades
- Modulized structures and configurations

  variations of Hamiltonian

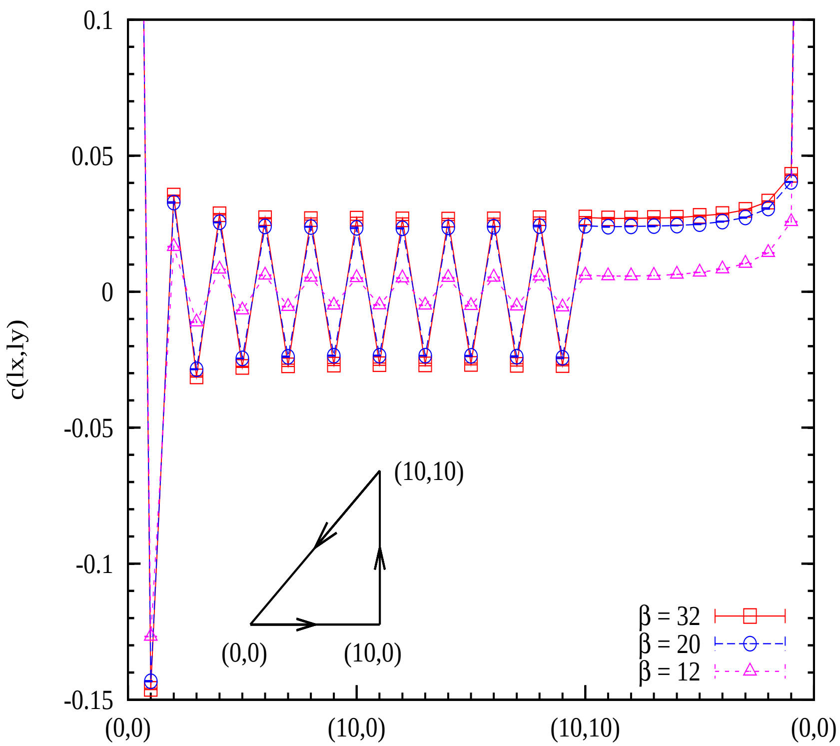
  different lattice geometry and multi-layer

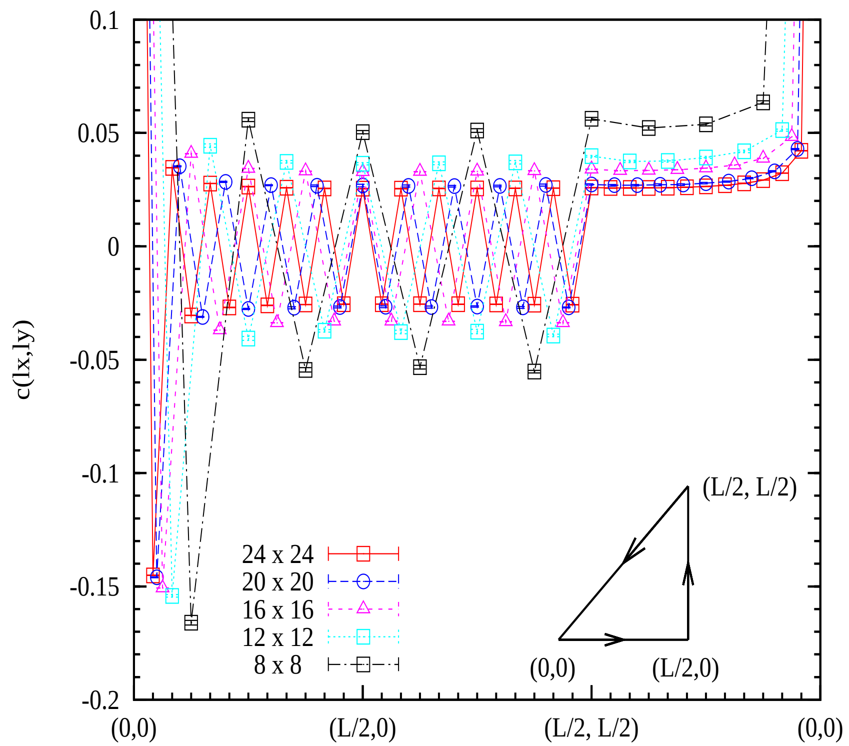
  many physical measurements
- *Stable and efficient multi-length scale matrix computation kernels*
- Partially parallelized (MPI, OpenMP)



QUEST

QUEST ⇒ “Right” Physics and “Record-breaking” lattice sizes





magnetism forms as T is lowered

large lattice sizes lead to converge

Matrix kernel

- Multi-length scale matrices

DQMC:

$$M_\sigma(h) = I + BD_L BD_{L-1} \cdots BD_1$$

HQMC:

$$M_\sigma(x) = I_{NL} - (I_N \otimes B)D_{[L]}(P \otimes I_N)$$

- $B = e^{t\Delta\tau K}$

$D_\ell = e^{\sigma V_\ell(x_\ell)}$

- $D_{[L]} = \text{diag}(D_1, D_2, \ldots, D_L)$

$V_\ell(x_\ell) = \text{diag}(x_1, x_2, \ldots, x_L)$

- K is defined based on lattice structure:

$K = K_x \oplus K_y$ for 2-D rectangle

Multi-length scaling

- Length-scales: N, L
  - N: spatial lattice size,
  - L: the number of imaginary-time slices,
- Energy scales: t, U, β
  - t: hopping of electrons between atoms and layers (kinetic energy),
  - U: strength of the interactions between the electrons (potential energy),
  - β: inverse temperature,
- Length and energy scale connection: $\Delta\tau = \frac{\beta}{L}$

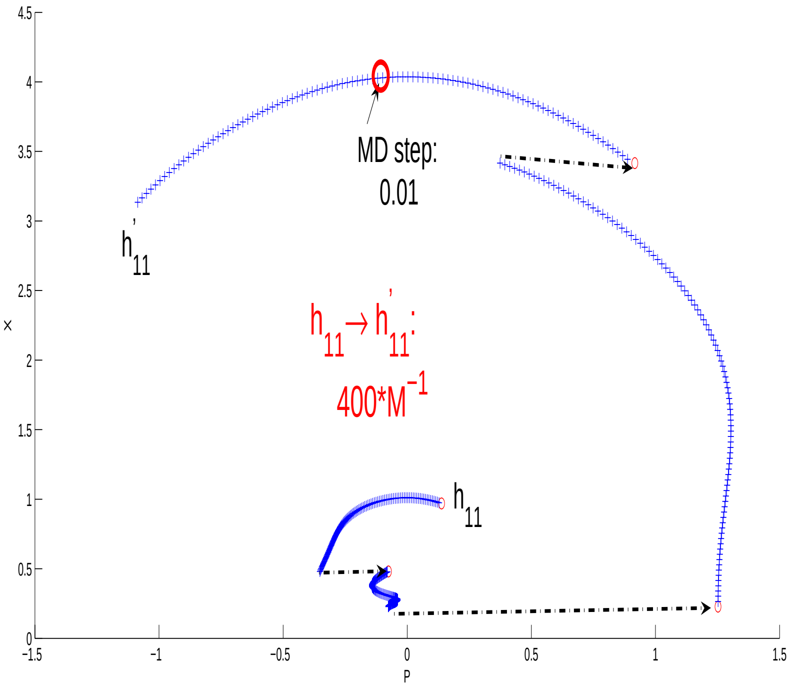
In more complex situations other energy scales also enter, such as the frequency of ionic vibrations (phonons) and the strength of the coupling of electrons to those vibrations

QMC simulation kernels

Matrix computation problems

$$\frac{\det[M_\sigma(h')]}{\det[M_\sigma(h)]}, \quad (M_\sigma^T(x)M_\sigma(x))^{-1}\Phi_\sigma, \quad G_\sigma(h) = M_\sigma^{-1}(h) \quad \ldots$$

A typical MD trajectory:



1. High order, say

$$\begin{aligned} N_x \times N_y \times L &= 64 \times 64 \times 160 \\ &= 655,360 \end{aligned}$$

2. Wide range of eigenvalue distributions $\lambda(M)$ and condition numbers $\kappa(M)$

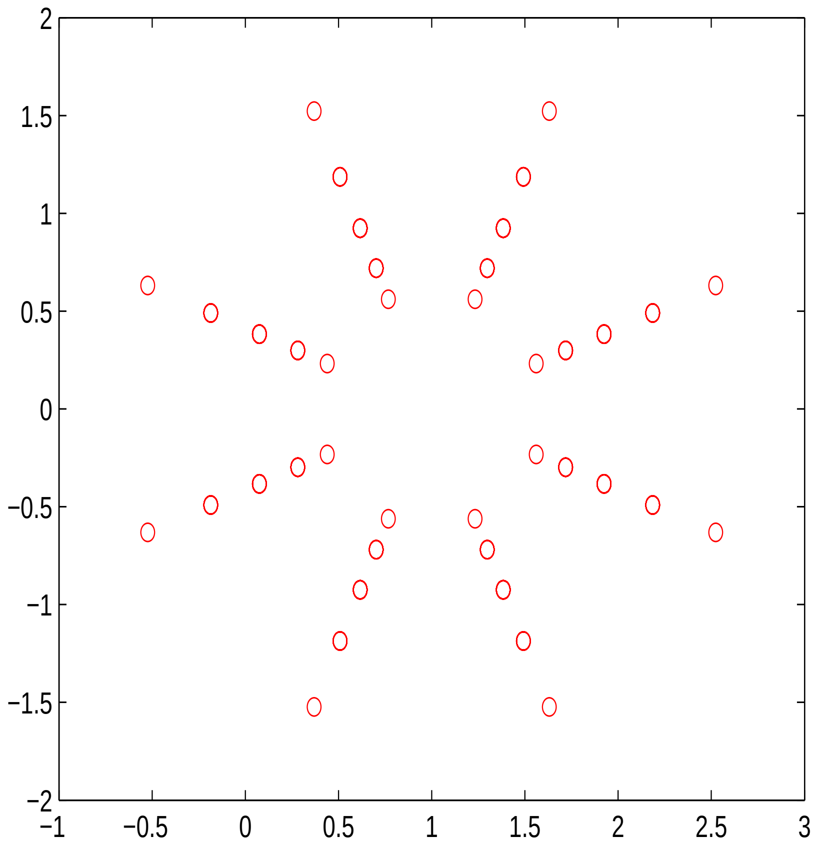
3. Metropolis Monte-Carlo, $\mathcal{O}(10^4)$ solutions required

Multi-length scale matrix computations

Hubbard matrix eigenvalue distribution $\lambda(M)$

$$\lambda(M) = 1 - \lambda(B_L \cdots B_2 B_1)^{\frac{1}{L}} e^{\mathtt{i}\frac{(2\ell+1)\pi}{L}}, \quad 0 \leq \ell \leq L-1$$

[Frobenius ’12, Romanovsky ’43, Varga ’62]

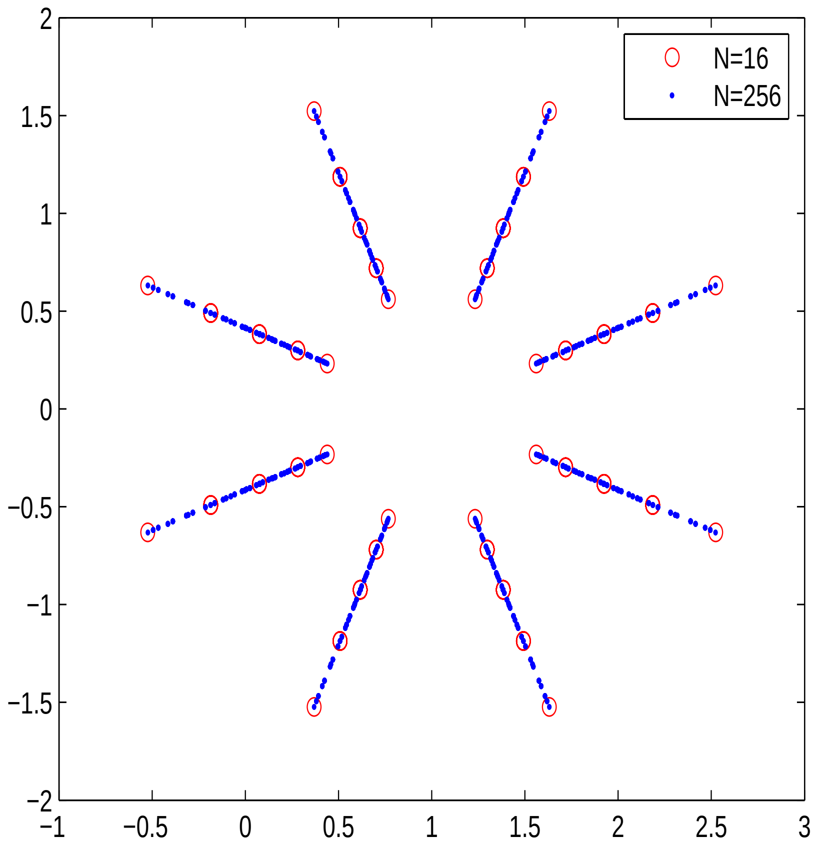


Multi-length scale matrix analysis

Hubbard matrix eigenvalue distribution $\lambda(M)$

$$\lambda(M) = 1 - \lambda(B_L \cdots B_2 B_1)^{\frac{1}{L}} e^{\mathtt{i}\frac{(2\ell+1)\pi}{L}}, \quad 0 \le \ell \le L-1$$

[Frobenius ’12, Romanovsky ’43, Varga ’62]

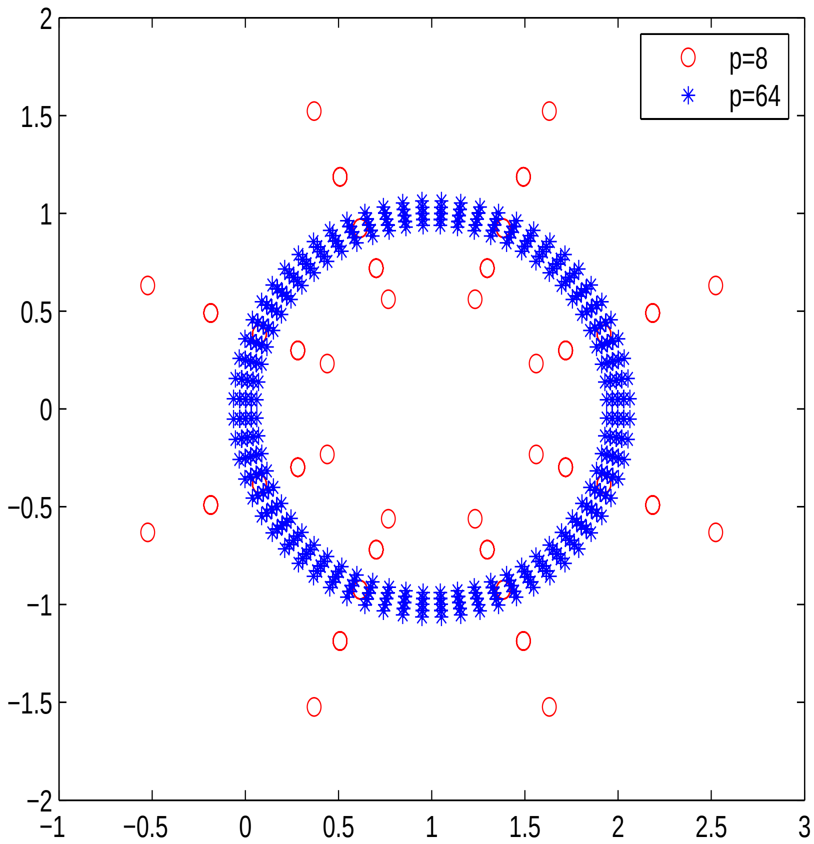


Multi-length scale matrix analysis

Hubbard matrix eigenvalue distribution $\lambda(M)$

$$\lambda(M) = 1 - \lambda(B_L \cdots B_2 B_1)^{\frac{1}{L}} e^{\mathrm{i}\frac{(2\ell+1)\pi}{L}}, \quad 0 \leq \ell \leq L-1$$

[Frobenius ’12, Romanovsky ’43, Varga ’62]

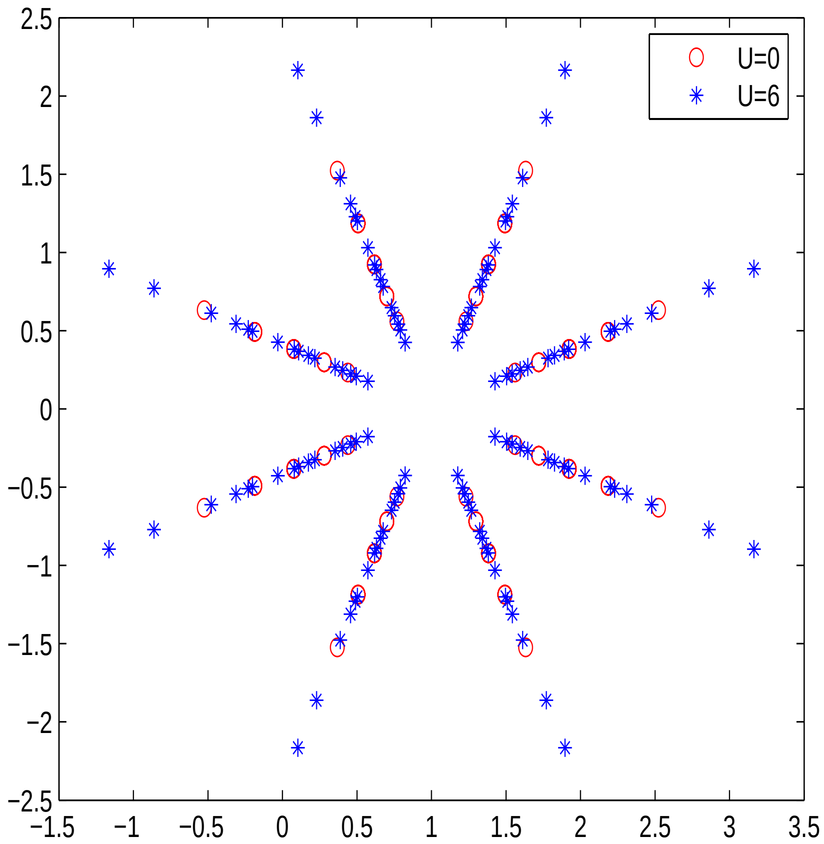


Multi-length scale matrix analysis

Hubbard matrix eigenvalue distribution $\lambda(M)$

$$\lambda(M) = 1 - \lambda(B_L \cdots B_2 B_1)^{\frac{1}{L}} e^{\mathtt{i}\frac{(2\ell+1)\pi}{L}}, \quad 0 \leq \ell \leq L-1$$

[Frobenius '12, Romanovsky '43, Varga '62]

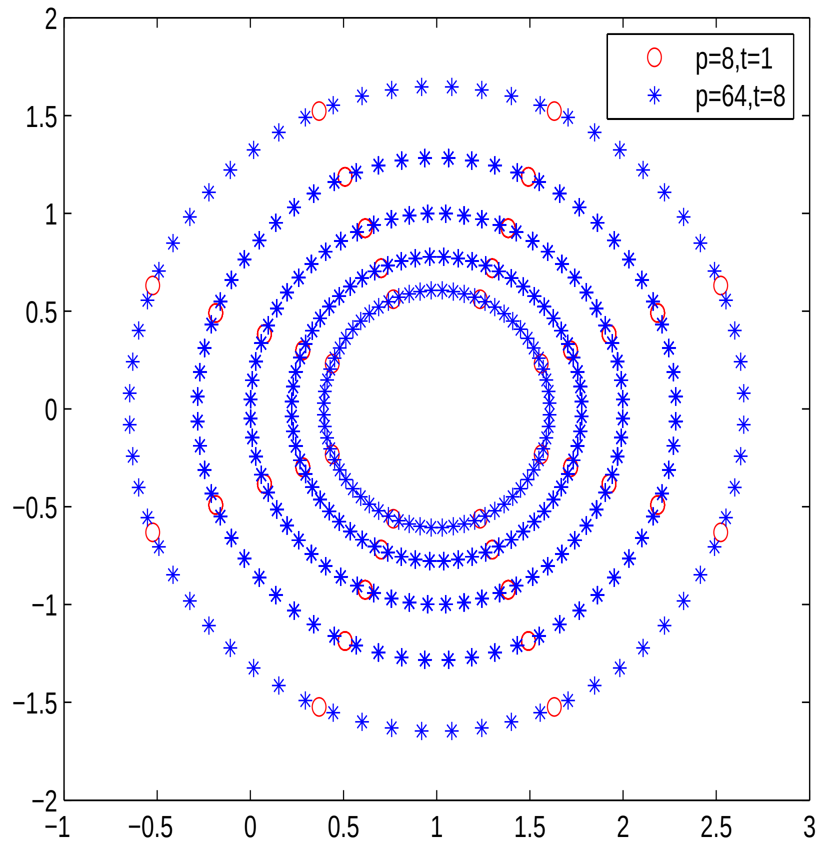


Multi-length scale matrix analysis

Hubbard matrix eigenvalue distribution $\lambda(M)$

$$\lambda(M) = 1 - \lambda(B_L \cdots B_2 B_1)^{\frac{1}{L}} e^{\mathrm{i}\frac{(2\ell+1)\pi}{L}}, \quad 0 \le \ell \le L-1$$

[Frobenius '12, Romanovsky '43, Varga '62]



Communication-avoiding stable matrix inversion

- Green's function

$$G = M^{-1} = (I + B_L B_{L-1} \cdots B_1)^{-1}$$

- Singular values of $B_L B_{L-1} \cdots B_1$ could spread out from 10^{30} to 10^{-30}
- A number of methods have been studied to calculate the matrix product in our community, Heath/Laub/Paige/Ward, Bojanczyk/Nagy/Plemmnons, van Dooren, Kagstrom, ...
- ***Graded (stratified)*** decomposition

$$B_L B_{L-1} \cdots B_1 = UDT = U \begin{bmatrix} \mathrm{X} & & & \\ & \mathrm{x} & & \\ & & \mathrm{x} & \\ & & & \mathrm{x} \end{bmatrix} T$$

  using QR with pivoting, and proper ordering of multiplications [Loh *et al*'92], or Jacobi rotations [Stewart'94].
- $G = M^{-1} = (I + UDT)^{-1} = T^{-1}(U^T T^{-1} + D)^{-1} U^T$

Communication-avoiding stable matrix inversion

- QR with pivoting is *not friendly* to multicore computing

BLAS/LAPACK in MKL on Intel Quad 2.4GHz

Gflops	`dgemm`	`dgeqrf`	`dgeqp3`
1-core	7.79	6.47	1.47
2-core	15.68	13.10	2.47
2-way Quad	26.39	21.68	3.22

- We developed an alternative based on a block structure orthogonal factorization (BSOF) *without pivoting*

	UDT	BSOF
1-core	4.02	7.68
2-core	5.06	13.28
2-way Quad	6.22	18.85

Sel-adapting direct linear solvers

- Block cyclic reduction

$$\begin{bmatrix} I & & & & B_1 \\ -B_2 & I & & & \\ & -B_3 & I & & \\ & & -B_4 & I & \\ & & & -B_5 & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix},$$

$$\Rightarrow \quad \begin{bmatrix} I & & B_1 \\ -B_3 B_2 & I & \\ & -B_5 B_4 & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_3^{(2)} \\ b_5^{(2)} \end{bmatrix},$$

- Factor of 2^k–reduction, however, limited k due to numerical instability

[Buzbee/Golub/Nielson'70, ...]

Sel-adapting direct linear solvers

- Block structured orthogonal factorization (BSOF)

$$M = (Q_1 Q_2 \cdots Q_L) R,$$

i.e.

$$\begin{bmatrix} I & & & B_1 \\ -B_2 & I & & \\ & \ddots & \ddots & \\ & & -B_L & I \end{bmatrix} \stackrel{Q_k}{\Longrightarrow} \begin{bmatrix} R_1 & X & & & X \\ & R_2 & X & & X \\ & & \ddots & \ddots & X \\ & & & R_{L-1} & X \\ & & & & R_L \end{bmatrix}.$$

- Rich substructure of $Q_1 Q_2 \cdots Q_L$ exploited for the Green's function calculations
- Parallelizable [Wright'92,...]
- Stable method, but, high memory cost $\mathcal{O}(N^2 L)$.

Block cyclic reduction + BSOF:

1. k-step block reduction:

$$Mx = b \quad \Longrightarrow \quad M^{(k)}x^{(k)} = b^{(k)}$$

   i.e.,

$$\text{block } L\text{–cyclic system} \quad \Longrightarrow \quad \text{block } \tfrac{L}{k}\text{–cyclic system}$$

2. BSOF

$$Q^T_{\frac{L}{k}-1} \cdots Q^T_1 M^{(k)} = R,$$

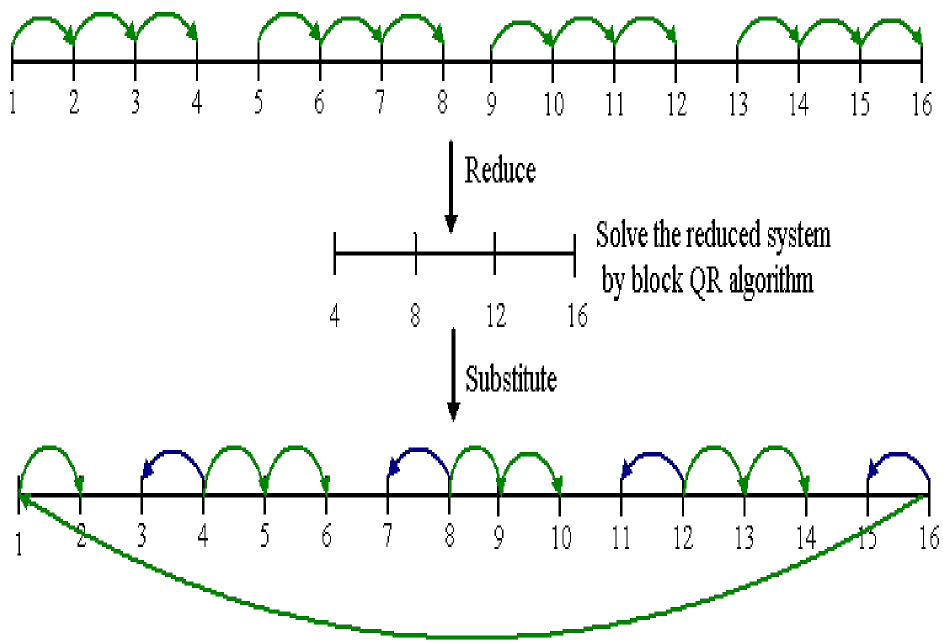
   and compute $x^{(k)}$

3. Forward and back substitutions:

$$x_i \quad \longleftarrow \quad x^{(k)} \quad \longrightarrow \quad x_j$$

Self-adapting direct linear solvers

Factor-k reduction:



- The order of $M^{(k)}$ is reduced by a factor of k.
- However, the condition number of $M^{(k)}$ increases when k increases.
- *How to* **self-adaptively** *determine the reduction factor* k *so that the computed solutions have the required accuracy?*

Self-adapting direct linear solvers

Self-adaptive reduction factor k

Given a desired accuracy

$$\frac{\|x_\ell - \widehat{x}_\ell\|}{\|x_\ell\|} \leq \text{tol},$$

by an error analysis of $\widehat{x}_\ell$ and conditioning estimation of $M^{(k)}$, the reduction factor k is then **adaptively** determined with respect to the simulation parameters $L(\beta), U, \ldots$:

$$k = \left\lfloor \frac{\frac{2}{3}\ln(\mathbf{tol}/\epsilon)}{4t\Delta\tau + \nu} \right\rfloor$$

where $\nu = \sqrt{U\Delta\tau} + \cdots$

Example: $t = 1$, $\Delta\tau = 1/8$, tol $= 10^{-8}$, $\epsilon = 10^{-16}$,

U	0	1	2	3	4	5	6
Reduction factor k	24	14	12	10	9	9	8

Robust preconditioned iterative linear solvers

- Preconditioned conjugate gradient (PCG)

$$M^T M x = b,$$

- Symmetrical preconditioned linear system

$$R^{-T}(M^T M)R^{-1} \cdot Rx = R^{-T}b,$$

- Earlier work on preconditioning techniques turned out to be of poor quality, and/or the growth of costs (memory and flops) significantlly as $N, U, \beta(L)$ increasing.
- Is there a linear scaling, $\mathcal{O}(NL)$, iterative solver?

Robust preconditioned iterative linear solvers

- Incomplete Cholesky (IC) factorization:

$$M^T M = R^T R + E,$$

  where R is an upper triangular matrix and E is the error matrix.

- In QMC simulation, it suffers
  - high cost to apply R due to large number of fill-ins,
  - or low quality (large number of iterations),
  - *not robust, pivot break-down due to loss of* $M^T M - E > 0$.

Robust preconditioned iterative linear solvers

- Robust Incomplete Cholesky (RIC)

$$\begin{cases} M^TM = R^TR + E \\ \text{subject to } \; E = R^TF + F^TR + S, \\ \qquad\qquad\quad M^TM - E > 0, \end{cases}$$

- *Robust, no pivot breakdown*
- Quality measured by the residual matrix

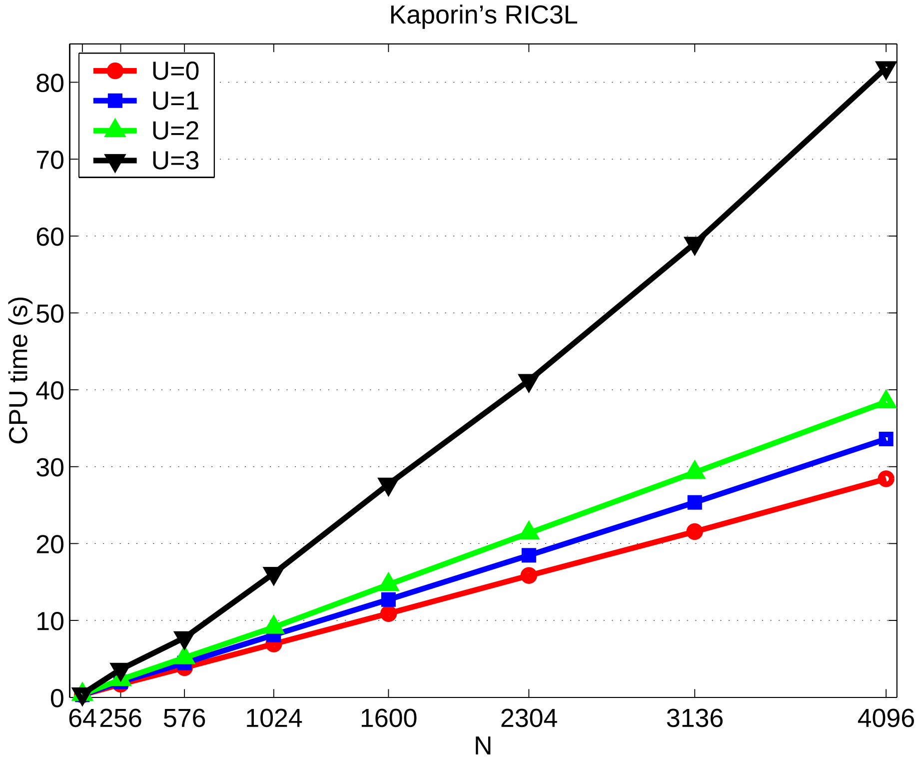
$$I - R^{-T}(M^TM)R^{-1} = \underbrace{FR^{-1} + R^{-T}F}_{\mathcal{O}(\|R^{-1}\|\,\sigma_1)} + \underbrace{R^{-T}SR^{-1}}_{\mathcal{O}(\|R^{-1}\|^2\sigma_2)},$$

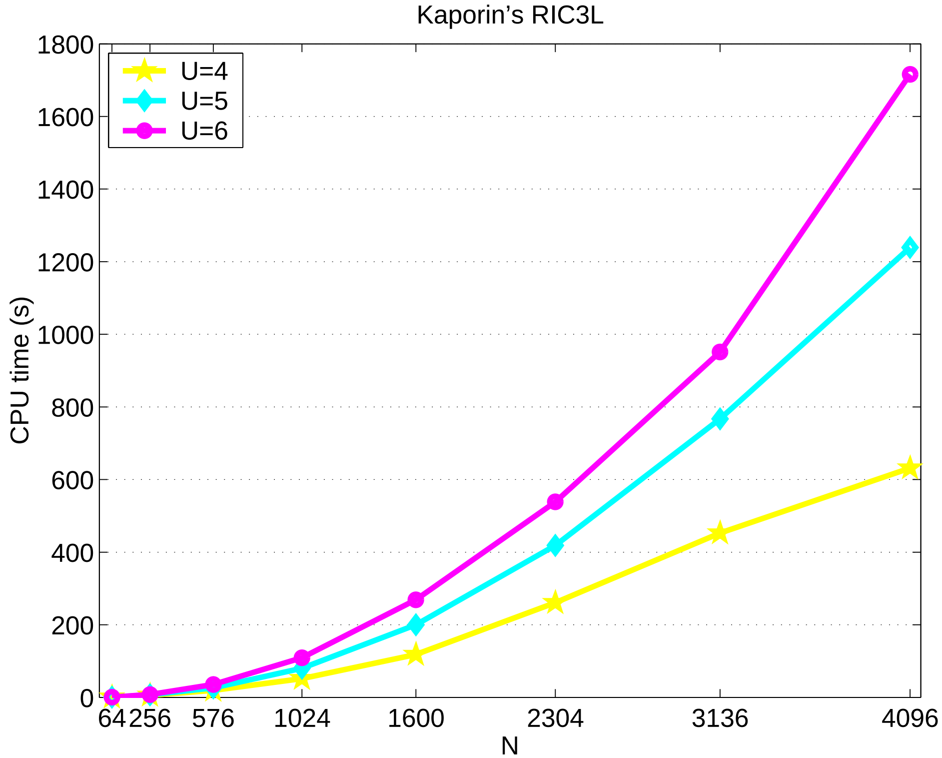
- Balance the cost and quality for multi-length scales using proper primary and secondary drop tolerances σ_1 and σ_2
- *An extended Compressed Sparse Column (CSC) storage format is proposed to accommodate the data access pattern.*
- Early work by Ajiz & Jennings, Tismenetsky, Kaporin, Benzi, and Tuma.
- I. Yamazaki's PhD thesis, 2008

Robust preconditioned iterative linear solvers

Good news: $\mathcal{O}(NL)$ for small U

Bad news: $\mathcal{O}(N^2L)$ for large U





Concluding remarks

1. Synergistic effort on the development of large-scale computational techniques for multi-length scale simulations in computational material science, where *"tensors run rampant!"*

2. Emerging opportunities for matrix/tensor research on
   - Robust and efficient algorithm design and analysis for multi-length scale matrices/tensors – *fully tensor-based?*
   - Structure exploitation
   - Multi-core matrix computing
   - Software and toolbox development

[B., Chen, Scalettar and Yamazaki: Lectre notes on numerical methods for quantum Monte Carlo simulations of the Hubbard model, $\sim$ 120 pages]