Motivation – Microscopic Models

- Microscopic Hamiltonians in second quantization
  e.g. Hubbard model
  \[ H = -t \sum_{\langle ij \rangle, \sigma} [c_{i\sigma}^\dagger c_{j\sigma} + H.c.] + U \sum_i n_{i\uparrow} n_{i\downarrow} \]

- e.g. Holstein-Hubbard model (HHM)
  \[ H = -t \sum_{\langle ij \rangle, \sigma} [c_{i\sigma}^\dagger c_{j\sigma} + H.c.] + U \sum_i n_{i\uparrow} n_{i\downarrow} + g \omega_0 \sum_{i,\sigma} (b_{i\sigma}^\dagger + b_{i\sigma}) n_{i\sigma} + \omega_0 \sum_i b_{i\uparrow}^\dagger b_{i\downarrow} \]

Hilbert space / #quantum states growth exponentially

HHM using an N-site lattice: \( 4^N \times (M+1)^N \) (N~10-100; M~10)

Electrons: Max. M per Site

Phonons: Max. M per Site
Motivation – Numerical Approaches to Ground-State Properties

Traditional Approaches
- Quantum Monte Carlo (QMC)
- Exact Diagonalization (ED): Massively Parallel Codes on Supercomputers

New Approach
- Density Matrix Renormalization Group (DMRG) Method
  - Originally introduced by White in 1992
  - Large sequential C++ package is in wide use (quantum physics and quantum chemistry)
  - Elapsed Times: hours to weeks with desktop CPUs
  - No parallel implementation available to date

DMRG Algorithm
- Basic Idea: Find an appropriate (reduced) basis set describing the ground-state of $H$ with high accuracy
- Basic Quantities:
  - Superblock = system & environment
  - Superblock state (product of system & environment states)
    \[ |\psi\rangle = \sum_{ij} \psi_{ij} |i\rangle |j\rangle \]
  - Reduced density matrix (DM): summation over environment states
    \[ \rho_{ii'} = \sum_j \psi_{ij}^* \psi_{i'j} \]
- Eigenstates of DM with largest eigenvalues have most impact on observables!
DMRG Algorithm

**DMRG algorithm (finite size; left to right sweep)**

1. Diagonalize the reduced DM for a system block of size $l$ and extract the $m$ eigenvectors with largest eigenvalue
2. Construct all relevant operators (system block & environment,...) for a system block of size $l+1$ in the reduced density matrix eigenbasis
3. Form a superblock Hamiltonian from system & environment Hamiltonians plus two single sites
4. Diagonalize the new superblock Hamiltonian

Accuracy depends mainly on $m$ ($m \sim 100 - 10000$)

---

**Implementation**

- **Start-Up with infinite-size algorithm**
- **DM diagonalization**: LAPACK (dsyev) costs about 5 %
- **Superblock diagonalization**: costs about 90 % (Davidson algorithm)
- Most time-consuming step: Sparse matrix-vector multiply (MVM) in Davidson (costs about 85 %)
- Sparse matrix $H$ is constructed by the transformations of each operator in $H$:

$$H_{i,j;i',j'} = \sum_\alpha A_{ii'}^\alpha B_{jj'}^\alpha$$

Contribution from **system block** and from **environment**
DMRG Algorithm: Parallelization

Implementation of sparse MVM

- Sparse MVM: Sum over dense matrix-matrix multiplies!
  \[ \sum_{i'j'} H_{i'j'} \psi_{i'j'} = \sum_{\alpha} \sum_{i'} A_{i'i}^{\alpha} \sum_{j'} B_{j'j}^{\alpha} \psi_{i'j'} \]

- However, \(A\) and \(B\) may contain only a few nonzero elements, e.g. if conservation laws (quantum numbers) have to be obeyed.

- To minimize overhead, an additional loop (running over nonzero blocks only) is introduced.

\[ H \psi = \sum_{\alpha} \sum_{k} (H \psi)_{L(k)}^{\alpha} \]
\[ = \sum_{\alpha} \sum_{k} A_{k}^{\alpha} \psi_{R(k)} \left[ B^{T} \right]_{k}^{\alpha} \]

DMRG Algorithm: Parallelization

Implementation of sparse MVM - pseudocode

```
// W: wavevector ; R: result
for (\(\alpha=0; \alpha < \text{number}\_\text{of}\_\text{hamiltonian}\_\text{terms}\); \(\alpha++\)) {
    term = \text{hamiltonian}\_\text{terms}[\alpha];
    for (\(k=0; k < \text{term}\_\text{number}\_\text{of}\_\text{blocks}\); \(k++\)) {
        li = term[k]\_\text{left}\_\text{index};
        ri = term[k]\_\text{right}\_\text{index};
        temp_matrix = term[k]\_\text{B}\_\text{transpose()} * W[ri];
        R[li] += term[k]\_A \* temp_matrix;
    }
}
```

Data dependency!

Matrix-Matrix-Multiply
(Parallel DGEMM ?!)
DMRG: OpenMP Parallelization

Implementation of parallel sparse MVM – pseudocode (main loop)

// W: wavevector ; R: result
#pragma omp parallel private(mymat, li, ri, myid, ics)
{
    myid = omp_get_thread_num();
    mytmat = mm[myid]; // temp thread local matrix
#pragma omp for
    for (ics=0; ics< icsmax; ics++) {
        li = block_array[ics]->left_index;
        ri = block_array[ics]->right_index;

        mytmat = block_array[ics]->B.transpose() * W[ri];

        omp_set_lock(locks[li]);
        R[li] += block_array[ics]->A * mytmat;
        omp_unset_lock(locks[li]);
    }
}

Fused (α,k) loop

Protect each block of result vector R with locks

Parallel DMRG: References


Parallel DMRG: Applications

- **Goals**
  - Show capabilities and limits of the parallel DMRG approach
    - Ground-state properties of 4x4 and 6x6 Hubbard systems at half-filling
  - Shed some light on issue of ground-state stripe formation in Hubbard ladders with open BCs in long (leg) direction and closed BCs in short (rung) direction
    - Stripes in ground-state hole density of 7x6, 11x6, 14x6 OxP with 4, 6 and 8 holes, respectively

Stripe Formation

4x4 and 6x6 periodic Hubbard lattices at half filling: $E/N$ vs. discarded weight

$E/N$ vs. discarded weight for 6x6 and 4x4 lattices.

7x6 OxP Hubbard ladder with 4 holes:

Ground-state energy vs. $m$

Transition at $m \approx 600$

W&S transition point at $m = 8000$

$U=12$
Strip Formation

7x6 OxP Hubbard ladder:

Ground-state energy vs. discarded weight

\[ E_g/N \]

\begin{align*}
2.0 \times 10^{-5} & \quad 4.0 \times 10^{-5} \\
6.0 \times 10^{-5} & \quad 8.0 \times 10^{-5} \\
1.0 \times 10^{-4} & \quad 2.0 \times 10^{-4}
\end{align*}

discarded weight

\begin{align*}
-0.62 & \quad -0.61 \\
-0.60 & \quad -0.59 \\
-0.58 & \quad -0.56
\end{align*}

7x6
11x6

y-integrated hole density

\begin{align*}
m=400 & \quad m=600 \\
m=800 & \quad m=1500 \\
m=2200 & \quad m=3600 \\
m=4500 & \quad m=6000
\end{align*}

transition at \( m \approx 600 \)

U=12
Reproduced W&S results for $U \in \{3,12\}$
- using reflection symmetry
- transition to striped state already at $m \approx 600$
- about 6 CPU weeks (Power4 1.3 GHz) for getting to $m = 8000$

What's next?
- Is stripe formation an artifact from the boundaries?
- Is stripe formation strongly dependent on the length of the ladder?
- Can we answer those questions by going to larger systems (in x direction) in order to shift the boundaries away from interesting structures?
- How far can we get with the current DMRG code?
11x6 OxP Hubbard ladder with 6 holes:
2D hole and spin densities

- Hole densities alone are not a reliable signature for stripe formation
- $\pi$ phase shifts of the magnetization density across the stripe provide additional evidence

Stripe signatures: Zero crossings of magnetization density (antiferromagnetic!)

$$S_x(x, y)(-1)^{x+y}$$

Symmetric vs. nonsymmetric calculation

Transition at $m \approx 1500$ for $U=12$
14x6 OxP Hubbard ladder with 8 holes: 2D hole and spin densities

Solid stripe formation!

Summary (1)

- Porting & parallelization of existing DMRG code from quantum physics/chemistry:
  - Kernel: sparse Matrix-Vector-Multiply (MVM)
  - Fusing inner & outer loop allows a scalable OpenMP implementation for MVM routine with a parallel efficiency of 98% for MVM
  - Good fraction of peak performance for whole application on modern SMP nodes

- Limits even of parallel DMRG show up clearly with periodic 6x6 Hubbard system at half filling
  - Convergence expected at $m \approx 10^5$
Summary (2)

- Reproduced 7x6 results by White & Scalapino
- Manageable system sizes for 6-leg OxP Hubbard ladders pushed as far as 21x6
  - Significantly earlier transition to striped state than W&S
  - Strange behaviour at 11x6, clear stripes at 14x6
  - 21x6 still inconclusive (lack of consistent theory)
- Using reflection symmetry is not always feasible
- Transition point to striped state is located at successively larger $m$ with increasing system size
- Solid signatures for convergence are still difficult to establish in DMRG