Parallel Coordinate Descent Methods for Full Configuration Interaction

Yuejia Zhang   Weiguo Gao   Yingzhou Li

Fudan University

June 2025

Problem: Searching for Ground-State

- The ground-state of a chemical system given by the many-body time-independent Schrödinger Equation

$$\hat{H}|\mathbf{\Phi_0}\rangle = E_0|\mathbf{\Phi_0}\rangle,$$

  where $|\mathbf{\Phi_0}\rangle = \mathbf{\Phi_0}(r_1, \ldots, r_{n_{\text{elec}}}), r_i \in \mathbb{R}^3$.
- Under Born–Oppenheimer approximation, the Hamiltonian operator with n_{nuc} nuclei and n_{elec} electrons is

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{n_{\text{elec}}} \nabla_i^2 + \sum_{i=1}^{n_{\text{elec}}} V_{\text{ext}}(r_i; \{R_I\}_{I=1}^{n_{\text{nuc}}}) + \sum_{i<j}^{n_{\text{elec}}} \frac{1}{\|r_i - r_j\|}.$$

FCI Numerical Discretization

- Schrödinger equation can be transformed to FCI eigenvalue problem

$$H\mathbf{c} = E_0\mathbf{c}, \quad H \in \mathbb{R}^{N_{\text{FCI}} \times N_{\text{FCI}}}, \quad \mathbf{c} \in \mathbb{R}^{N_{\text{FCI}}},$$

but ...

Table: Different Molecule Systems and Storage cost

Molecule	Basis	Electrons	Spin–Orbitals	Dimension	Memory
H_2O	cc-pVDZ	10	48	$\sim 10^8$	$\sim$ 1 GB
N_2	cc-pVDZ	14	56	$\sim 10^{11}$	$\sim$ 1 TB
N_2	cc-pVTZ	14	120	$\sim 10^{16}$	$\sim$ 100 PB
Cr_2	Ahlrichs	48	84	$\sim 10^{22}$	-

Why is FCI so Huge?

- Electrons fill orbitals: one-electron spin-orbitals $\{\chi_p\}_{p=1}^{n_{\text{orb}}}$ from Hartree–Fock procedure.
- Each configuration is a *Slater determinant* (or, *anti-symmetrized tensor products*):

$$|D_i\rangle = |\chi_{p_1}\chi_{p_2}\cdots\chi_{p_n}\rangle\,,$$

  with $n = n_{\text{elec}}$ and $1 \le p_1 < p_2 < \cdots < p_n \le n_{\text{orb}}$.
- FCI variational space dimension: $N_{\text{FCI}} = \binom{n_{\text{orb}}}{n_{\text{elec}}}$. Combinatorial explosion!
- Traditional methods fail due to dimensionality.

Hamiltonian Matrix

Entry: $H_{ij} = \langle D_i | \hat{H} | D_j \rangle$, not guaranteed to be non-negative.

- Symmetric. $H_{ij} = H_{ji}$.
- Sparse. For off-diagonals $|D_i\rangle \neq |D_j\rangle$,
  - If $|D_i\rangle = a_r^\dagger a_p |D_j\rangle$, $H_{ij} = \langle r|\hat{h}|p\rangle + \sum_k \langle rk||pk\rangle$.
  - If $|D_i\rangle = a_r^\dagger a_s^\dagger a_p a_q |D_j\rangle$, $H_{ij} = \langle rs||pq\rangle$.
  - Otherwise, $H_{ij} = 0$.

  Consequence: H has $O(n_{\text{elec}}^2 n_{\text{orb}}^2)$ entries per row.
- Ground-state eigenvalue $E_0 < 0$.
- Ground-state eigenvector **c** sparse in the sense of truncation.

Existing Methods

- **Selected CI**: Pick the important configurations.
  - Adaptive space growth based on energy estimates.
  - Good for weakly correlated systems, cuts down cost with little accuracy loss.
- **DMRG**: Tensor compression.
  - Wavefunction as matrix product state (MPS), compact storage.
  - Works best when entanglement is low (e.g., 1D-like systems).
- **FCIQMC**: Stochastic sampling in configuration space.
  - Monte Carlo walkers project onto the ground state.
  - Saves memory, but noisy; needs enough walkers to stabilize.

Coordinate Descent FCI (CDFCI)[1]

Coordinate gradient descent method

- Minimizes computational costs by *avoiding operations with the entire Hamiltonian matrix*.
- In each iteration, only one coordinate of the optimizing vector is updated.
- Computation for updating involves only one column of the Hamiltonian matrix.

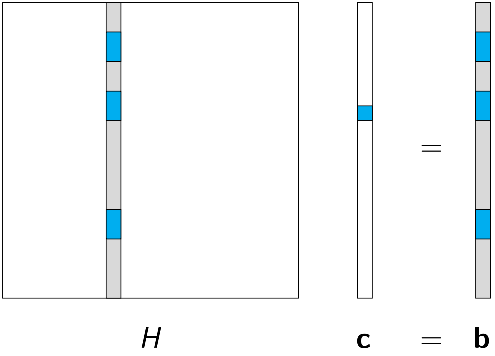


Figure: Update for one coordinate.

[1]Z. Wang, Y. Li, J. Lu, J. Chem. Theory Comput., 2019.

Solving Leading Eigenvalue Problem

Consider the unconstrained minimization problem

$$\min_{\mathbf{c}\in\mathbb{R}^{N_{\mathrm{FCI}}}} f(\mathbf{c}) = \min_{\mathbf{c}} \|H + \mathbf{c}\mathbf{c}^{\mathsf{T}}\|_F^2.$$

- Gradient $\nabla f(c) = 4H\mathbf{c} + 4(\mathbf{c}^{\mathsf{T}}\mathbf{c})\mathbf{c}$.
- Hessian $\nabla^2 f(c) = 4H + 8\mathbf{c}\mathbf{c}^{\mathsf{T}} + 4(\mathbf{c}^{\mathsf{T}}\mathbf{c})I$.
- Non-convex problem, with unbounded Lipschitz constant.
- Stationary points: $0, \pm\sqrt{-\lambda_1}\mathbf{v}_1, \ldots, \pm\sqrt{-\lambda_m}\mathbf{v}_m$ $(\cdots < \lambda_m < 0 < \lambda_{m+1} < \cdots)$.
- Only two local minimizers $\pm\sqrt{-\lambda_1}\mathbf{v}_1$ (which are also global minimizers), the others are all strict saddle points.
- Ensures convergence to the ground state $\pm\mathbf{c}$, given a good starting point (e.g., Hartree–Fock ground state).

CDFCI Framework

Initialize $\mathbf{c}^{(0)}$, $\mathbf{b}^{(0)} = H\mathbf{c}^{(0)}$.

For iteration $\ell = 1, 2, \ldots$

1. Select coordinate
   $i^{(\ell)} = \arg\max_i |\nabla_i f(\mathbf{c}^{(\ell-1)})|$.
2. Find stepsize by exact line search
   $\alpha^{(\ell)} = \arg\min_\alpha f(\mathbf{c}^{(\ell-1)} + \alpha \mathbf{e}_{i^{(\ell)}})$.
3. Update $\mathbf{c}^{(\ell)} = \mathbf{c}^{(\ell-1)} + \alpha^{(\ell)} \mathbf{e}_{i^{(\ell)}}$,
   $\mathbf{b}^{(\ell)} = \mathbf{b}^{(\ell-1)} + \alpha^{(\ell)} H_{:,i^{(\ell)}}$.

Remark: Gradient

$\nabla f(\mathbf{c}) = 4H\mathbf{c} + 4\mathbf{c}^\mathsf{T}\mathbf{c}\mathbf{c} = 4\mathbf{b} + 4\mathbf{c}^\mathsf{T}\mathbf{c}\mathbf{c}$.

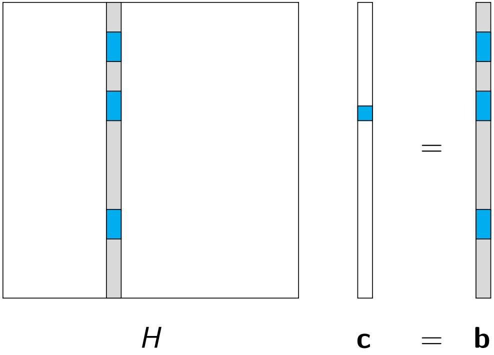


Figure: Update for one coordinate.

CDFCI Framework - for Two Coordinates?

Initialize $\mathbf{c}^{(0)}$, $\mathbf{b}^{(0)} = H\mathbf{c}^{(0)}$.

For iteration $\ell = 1, 2, \ldots$

1. Select coordinate
   $i^{(\ell)} = \arg\max_i |\nabla_i f(\mathbf{c}^{(\ell-1)})|$,
   $j^{(\ell)} = \arg\max_{j \neq i^{(\ell)}} |\nabla_j f(\mathbf{c}^{(\ell-1)})|$.
2. Find stepsize
   $\alpha^{(\ell)} = \arg\min_\alpha f(\mathbf{c}^{(\ell-1)} + \alpha \mathbf{e}_{i^{(\ell)}})$,
   $\beta^{(\ell)} = \arg\min_\beta f(\mathbf{c}^{(\ell-1)} + \beta \mathbf{e}_{j^{(\ell)}})$.
3. Update
   $\mathbf{c}^{(\ell)} = \mathbf{c}^{(\ell-1)} + \alpha^{(\ell)} \mathbf{e}_{i^{(\ell)}} + \beta^{(\ell)} \mathbf{e}_{j^{(\ell)}}$,
   $\mathbf{b}^{(\ell)} = \mathbf{b}^{(\ell-1)} + \alpha^{(\ell)} H_{:,i^{(\ell)}} + \beta^{(\ell)} H_{:,j^{(\ell)}}$.

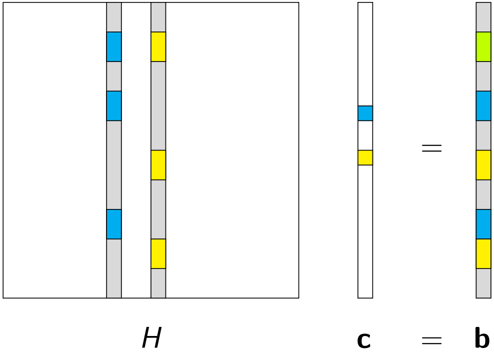


Figure: Update for two coordinates.

CDFCI Framework - Exact Line Search?

Initialize $\mathbf{c}^{(0)}$, $\mathbf{b}^{(0)} = H\mathbf{c}^{(0)}$.

For iteration $\ell = 1, 2, \ldots$

1. Select coordinate
   $i^{(\ell)} = \arg\max_i |\nabla_i f(\mathbf{c}^{(\ell-1)})|$,
   $j^{(\ell)} = \arg\max_{j \neq i^{(\ell)}} |\nabla_j f(\mathbf{c}^{(\ell-1)})|$.
2. Find stepsize $\alpha^{(\ell)}, \beta^{(\ell)} = \arg\min_{\alpha,\beta} f(\mathbf{c}^{(\ell-1)} + \alpha \mathbf{e}_{i^{(\ell)}} + \beta \mathbf{e}_{j^{(\ell)}})$.
3. Update
   $\mathbf{c}^{(\ell)} = \mathbf{c}^{(\ell-1)} + \alpha^{(\ell)} \mathbf{e}_{i^{(\ell)}} + \beta^{(\ell)} \mathbf{e}_{j^{(\ell)}}$,
   $\mathbf{b}^{(\ell)} = \mathbf{b}^{(\ell-1)} + \alpha^{(\ell)} H_{:,i^{(\ell)}} + \beta^{(\ell)} H_{:,j^{(\ell)}}$.

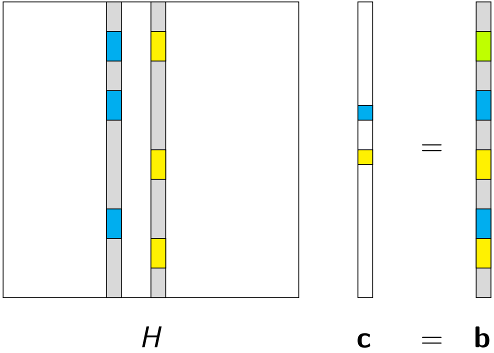


Figure: Update for two coordinates.

Add a Scalar γ for Exact Line Search

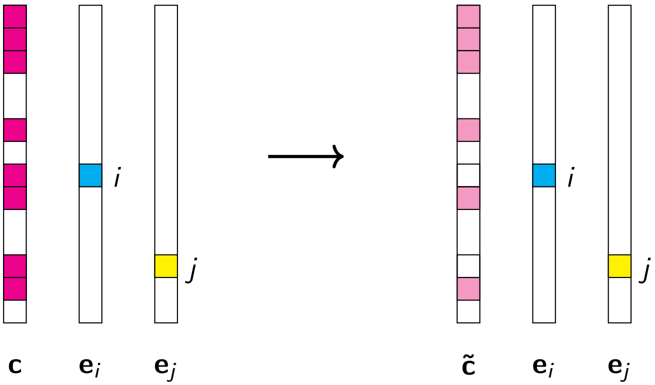
Modify the minimization problem from $\min_{\alpha,\beta} f(\mathbf{c} + \alpha\mathbf{e}_i + \beta\mathbf{e}_j)$ to

$$\min_{\gamma,\alpha,\beta} f(\gamma\mathbf{c} + \alpha\mathbf{e}_i + \beta\mathbf{e}_j) = f(\begin{bmatrix}\mathbf{c} & \mathbf{e}_i & \mathbf{e}_j\end{bmatrix}\begin{bmatrix}\gamma\\ \alpha\\ \beta\end{bmatrix})$$

$$= \left\| H + \begin{bmatrix}\mathbf{c} & \mathbf{e}_i & \mathbf{e}_j\end{bmatrix}\begin{bmatrix}\gamma\\ \alpha\\ \beta\end{bmatrix}\begin{bmatrix}\gamma & \alpha & \beta\end{bmatrix}\begin{bmatrix}\mathbf{c}^T\\ \mathbf{e}_i^T\\ \mathbf{e}_j^T\end{bmatrix}\right\|_F^2 .$$

Matrix Orthogonalization

Construct $\begin{bmatrix} \tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j \end{bmatrix}$, where $\|\tilde{\mathbf{c}}\|_2 = 1$, $(\tilde{\mathbf{c}}, \mathbf{e}_i) = 0$, $(\tilde{\mathbf{c}}, \mathbf{e}_j) = 0$.



Add γ and $\tilde{\mathbf{c}}$ for Exact Line Search

Modify the minimization problem from $\min_{\alpha,\beta} f(\mathbf{c} + \alpha\mathbf{e}_i + \beta\mathbf{e}_j)$ to

$$\min_{\gamma,\alpha,\beta} f(\gamma\tilde{\mathbf{c}} + \alpha\mathbf{e}_i + \beta\mathbf{e}_j) = f\left(\begin{bmatrix}\tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j\end{bmatrix}\begin{bmatrix}\gamma \\ \alpha \\ \beta\end{bmatrix}\right)$$

$$= \left\| H + \begin{bmatrix}\tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j\end{bmatrix}\begin{bmatrix}\gamma \\ \alpha \\ \beta\end{bmatrix}\begin{bmatrix}\gamma & \alpha & \beta\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{c}}^T \\ \mathbf{e}_i^T \\ \mathbf{e}_j^T\end{bmatrix} \right\|_F^2$$

$$= \left\| \underbrace{\begin{bmatrix}\tilde{\mathbf{c}}^T \\ \mathbf{e}_i^T \\ \mathbf{e}_j^T\end{bmatrix} H \begin{bmatrix}\tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j\end{bmatrix}}_{\in \mathbb{R}^{3\times 3}} + \begin{bmatrix}\gamma \\ \alpha \\ \beta\end{bmatrix}\begin{bmatrix}\gamma & \alpha & \beta\end{bmatrix} \right\|_F^2 + \text{constant}.$$

Extension to Multi Coordinate Descent FCI

- Select a set of coordinates $I = \{i_1, \ldots, i_k\}, 1 \leq i_j \leq N_{\mathrm{FCI}}$ based on gradient $\nabla f(\mathbf{c}) = 4H\mathbf{c} + 4\mathbf{c}^{\mathrm{T}}\mathbf{c}\mathbf{c}$.
- Denote $\mathcal{E}_I = [e_{i_1}, \ldots, e_{i_k}] \in \mathbb{R}^{N_{\mathrm{FCI}} \times k}$.
- The update is given by
$$\mathbf{c} \leftarrow \gamma\mathbf{c} + \mathcal{E}_I\mathbf{a}.$$
- The values of γ and $\mathbf{a}$ are given by the eigenvector of
$$\begin{bmatrix} \tilde{\mathbf{c}}^{\mathrm{T}} \\ \mathcal{E}_I^{\mathrm{T}} \end{bmatrix} H \begin{bmatrix} \tilde{\mathbf{c}} & \mathcal{E}_I \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}$$
corresponding to the minimal eigenvalue $\lambda_{\min}$, which is the current energy estimate.

Compression Strategy

- Compression occurs when updating $\mathbf{b} = H\mathbf{c}$:

$$\mathbf{b} \leftarrow \gamma \mathbf{b} + H\mathcal{E}_I \mathbf{a},$$

$$b_i \leftarrow \gamma b_i + \sum_{j \in I} H_{ij} a_j.$$

- Update $H_{ij} a_j$ is discarded if $\mathbf{c}_i = 0$ and $|H_{ij} a_j| < \tau$.
- Not affecting eigenvalue estimator

$$\mathrm{RQ}(\mathbf{c}) = \frac{\mathbf{c}^\mathsf{T} H \mathbf{c}}{\mathbf{c}^\mathsf{T} \mathbf{c}} = \frac{\mathbf{c}^\mathsf{T} \mathbf{b}}{\mathbf{c}^\mathsf{T} \mathbf{c}}.$$

- Tolerance τ balances between memory-cost and accuracy.

Parallel Implementation Details

- Shared memory parallelism based on OpenMP.
- Sparse vectors $\mathbf{c}$ and $\mathbf{b} = H\mathbf{c}$ are stored in a concurrent hash table[2] that supports parallel read/write.
- Two levels of parallelism: row-wise and column-wise.

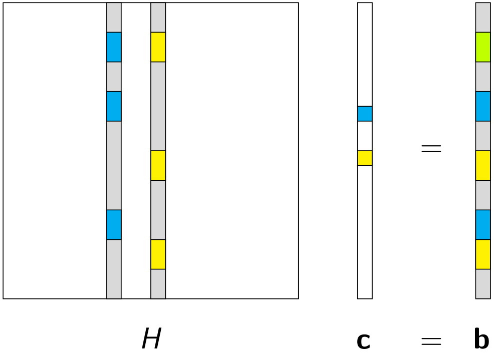


Figure: Update for two coordinates.

[2] https://github.com/efficient/libcuckoo

Overall Speedup: H_2O/cc-pVDZ

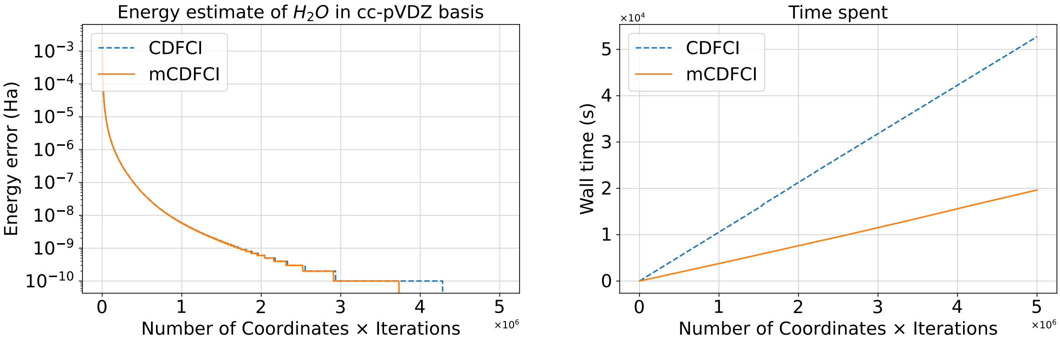
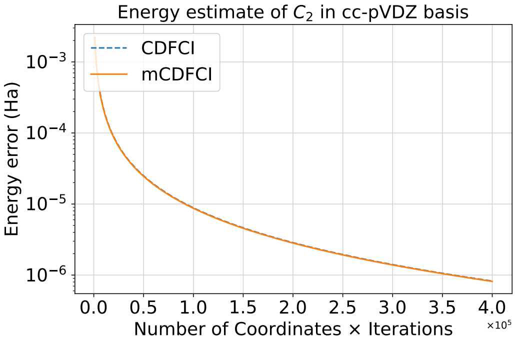


Figure: Speedup of mCDFCI compared with CDFCI, both in 64 threads and $k = 64$.

Overall Speedup: C_2/cc-pVDZ



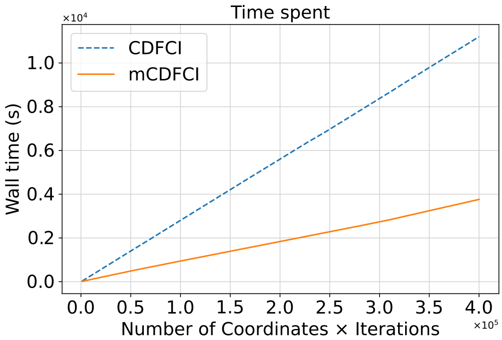
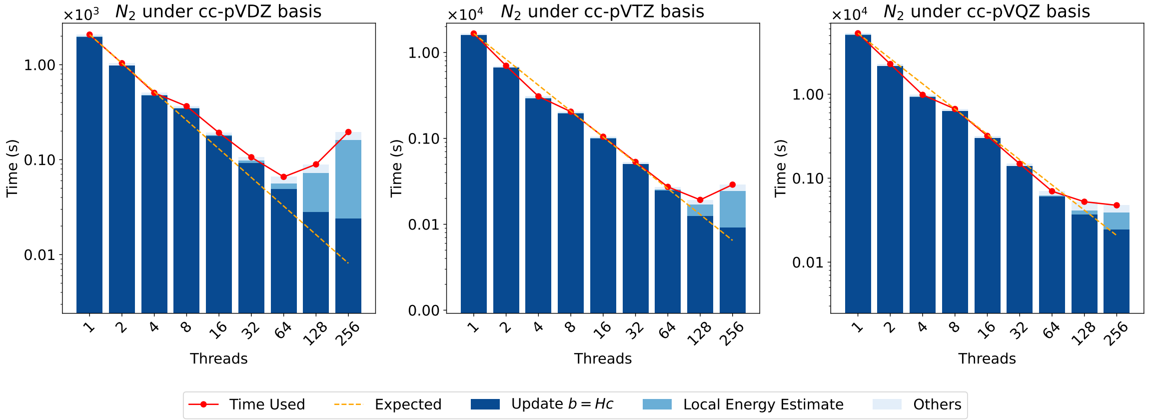


Figure: Speedup of mCDFCI compared with CDFCI, both in 128 threads and $k = 128$.

Compare with Other FCI Methods

Algorithm	Parameter	Absolute Error	Mem(GB)	Time(s)	mCDFCI Time(s)	mCDFCI Ratio
SHCI	$\varepsilon_1 = 1.0 \times 10^{-4}$	1.5×10^{-3}	16.1	3.3	14.0	0.23×
	$\varepsilon_1 = 5.0 \times 10^{-5}$	7.6×10^{-4}	17.5	7.3	23.7	0.31×
	$\varepsilon_1 = 1.0 \times 10^{-5}$	1.1×10^{-4}	32.6	53.9	113.2	0.48×
	$\varepsilon_1 = 5.0 \times 10^{-6}$	4.6×10^{-5}	54.2	126.0	219.4	0.57×
	$\varepsilon_1 = 1.0 \times 10^{-6}$	4.7×10^{-6}	238.1	845.0	954.1	0.89×
DMRG	$\max M = 500$	6.9×10^{-4}	0.2	106.9	23.7	4.52×
	$\max M = 1000$	1.5×10^{-4}	0.8	345.2	92.7	3.72×
	$\max M = 2000$	1.9×10^{-5}	3.1	1130.1	394.9	2.86×
	$\max M = 4000$	2.1×10^{-6}	11.5	3848.3	1549.2	2.48×
iS-FCIQMC	$m = 10000$	$7.4 \pm 1.6 \times 10^{-4}$	0.076	19.3	23.7	0.81×
	$m = 50000$	$6.4 \pm 0.6 \times 10^{-4}$	0.077	59.8	26.9	2.22×
	$m = 100000$	$1.8 \pm 0.4 \times 10^{-4}$	0.077	118.0	78.6	1.50×
	$m = 500000$	$2.3 \pm 0.2 \times 10^{-4}$	0.080	459.9	64.5	7.13×
	$m = 1000000$	$5.4 \pm 1.2 \times 10^{-5}$	0.083	872.4	196.5	4.44×

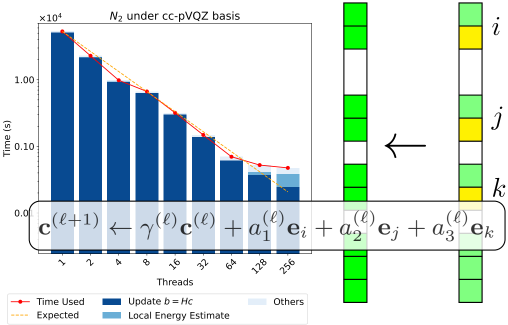
Scalability



Summary

The proposed method **mCDFCI**

- performs configuration selection using coordinate descent and exact line search.
- visits important determinants efficiently.
- captures the significant part of FCI space for ground state approximation.



Thanks for Your Attention!

Contact: yuejiazhang21@m.fudan.edu.cn
Preprint: https://arxiv.org/abs/2411.07565