Parallel Coordinate Descent Methods for Full Configuration Interaction

Yuejia Zhang   Weiguo Gao   Yingzhou Li

Fudan University, China

GAMM 2024, Magdeburg, Germany

Problem Statement

- Search for 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 $|\mathbf{\Phi_0}\rangle = \mathbf{\Phi_0}(r_1, \dots, r_{n_{\text{elec}}}), r_i \in \mathbb{R}^3$.

- 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

- Based on one-electron spin–orbitals $\{\chi_p\}_{p=1}^{n_{\text{orb}}}$ from Hartree–Fock procedure.
- Wavefunction approximated as linear combination of anti-symmetrized tensor products (Slater determinants)

$$|\mathbf{\Phi_0}\rangle = \sum_{i=1}^{N_{\text{FCI}}} c_i |D_i\rangle = \sum_{i=1}^{N_{\text{FCI}}} c_i |\chi_{p_i}\chi_{p_j}\cdots\chi_{p_k}\rangle.$$

- FCI variational space dimension: $N_{\text{FCI}} = \binom{n_{\text{orb}}}{n_{\text{elec}}}$.
- Schrödinger equation transformed to FCI eigenvalue problem

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

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 $O(n_{\mathrm{elec}}^2 n_{\mathrm{orb}}^2)$ entries per row.
- Ground-state eigenvalue $E_0 < 0$.
- Ground-state eigenvector $\mathbf{c}$ sparse in the sense of truncation.

Memory usage

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	$\sim 10^{22}$	-

Solution: Wavefunction Compression

- By tensor train: DMRG
- By sampling: FCIQMC, iFCIQMC, S-FCIQMC
- By selecting "important" configurations: HCI, SHCI, ASCI, **CDFCI**

FCI eigenvalue problem

Consider the unconstrained minimization problem

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

- 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 constraint.
- Stationary points: $0, \pm\sqrt{-\lambda_1}\mathbf{v}_1, \dots, \pm\sqrt{-\lambda_m}\mathbf{v}_m$ $(\cdots < \lambda_m < 0 < \lambda_{m+1} < \cdots)$.
- 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).

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.

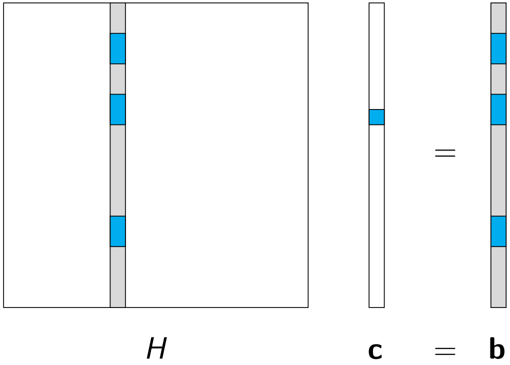


Figure: Update for one coordinate.

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

CDFCI Framework

Initialize $\mathbf{c}^{(0)}$, $\mathbf{b}^{(0)} = H\mathbf{c}^{(0)}$.
For iteration $\ell = 1, 2, \ldots$

1. Select coordinate
   $i^{(\ell)} = \arg\max_i \left|\nabla_i f(\mathbf{c}^{(\ell-1)})\right|$.
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
$\nabla f(\mathbf{c}) = 4H\mathbf{c} + 4\mathbf{c}^\mathsf{T}\mathbf{c}\mathbf{c} = 4\mathbf{b} + 4\mathbf{c}^\mathsf{T}\mathbf{c}\mathbf{c}$.

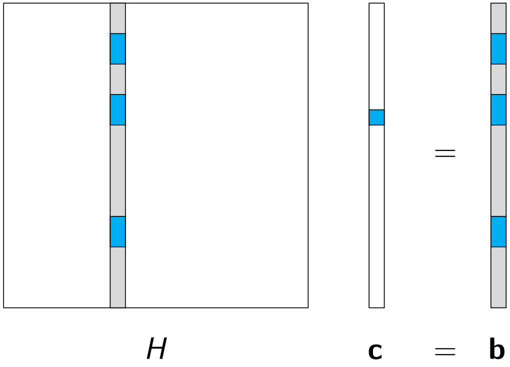


Figure: Update for one coordinate.

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
   $\mathbf{c}^{(\ell)} = \mathbf{c}^{(\ell-1)} + \alpha^{(\ell)} \mathbf{e}_{i^{(\ell)}} + \beta^{(\ell)} \mathbf{e}_{j^{(\ell)}}$,
   $\mathbf{b}^{(\ell)} = \mathbf{b}^{(\ell-1)} + \alpha^{(\ell)} H_{:,i^{(\ell)}} + \beta^{(\ell)} H_{:,j^{(\ell)}}$.

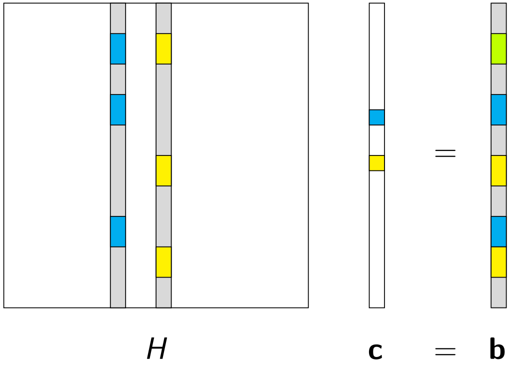


Figure: Update for two coordinates.

CDFCI Framework - Exact Line Search?

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

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

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
   $\mathbf{c}^{(\ell)} = \mathbf{c}^{(\ell-1)} + \alpha^{(\ell)} \mathbf{e}_{i^{(\ell)}} + \beta^{(\ell)} \mathbf{e}_{j^{(\ell)}}$,
   $\mathbf{b}^{(\ell)} = \mathbf{b}^{(\ell-1)} + \alpha^{(\ell)} H_{:,i^{(\ell)}} + \beta^{(\ell)} H_{:,j^{(\ell)}}$.

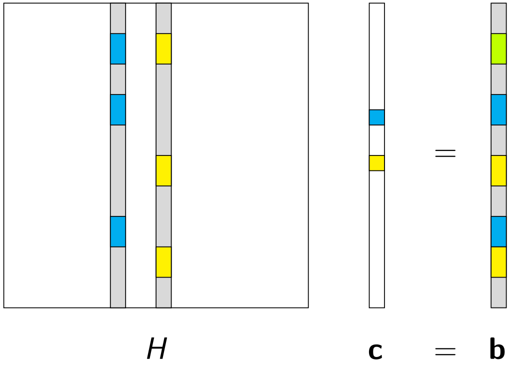


Figure: Update for two coordinates.

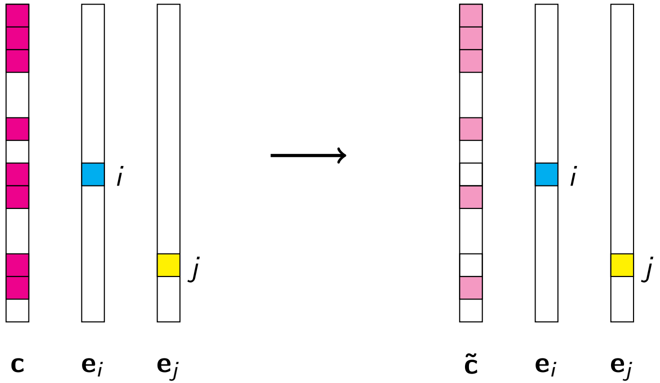
Add a Scalar γ for Exact Line Search

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

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

Matrix Orthogonalization

Construct $\begin{bmatrix} \color{red}{\tilde{\mathbf{c}}} & \mathbf{e}_i & \mathbf{e}_j \end{bmatrix}$, 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

$$\begin{aligned}
\min_{\gamma,\alpha,\beta} f(\gamma \tilde{\mathbf{c}} + \alpha \mathbf{e}_i + \beta \mathbf{e}_j) &= f\left(\begin{bmatrix} \tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j \end{bmatrix} \begin{bmatrix} \gamma \\ \alpha \\ \beta \end{bmatrix}\right) \\
&= \left\| H + \begin{bmatrix} \tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j \end{bmatrix} \begin{bmatrix} \gamma \\ \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \gamma & \alpha & \beta \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{c}}^{\mathsf{T}} \\ \mathbf{e}_i^{\mathsf{T}} \\ \mathbf{e}_j^{\mathsf{T}} \end{bmatrix} \right\|_F^2 \\
&= \left\| \underbrace{\begin{bmatrix} \tilde{\mathbf{c}}^{\mathsf{T}} \\ \mathbf{e}_i^{\mathsf{T}} \\ \mathbf{e}_j^{\mathsf{T}} \end{bmatrix} H \begin{bmatrix} \tilde{\mathbf{c}} & \mathbf{e}_i & \mathbf{e}_j \end{bmatrix}}_{\in \mathbb{R}^{3\times 3}} + \begin{bmatrix} \gamma \\ \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \gamma & \alpha & \beta \end{bmatrix} \right\|_F^2 .
\end{aligned}$$

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

$$\begin{bmatrix} \tilde{\mathbf{c}}^{\mathsf{T}} \\ \mathcal{E}_I^{\mathsf{T}} \end{bmatrix} H \begin{bmatrix} \tilde{\mathbf{c}} & \mathcal{E}_I \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}$$

corresponding to the minimal eigenvalue $\lambda_{\min}$, which is the current energy estimate.

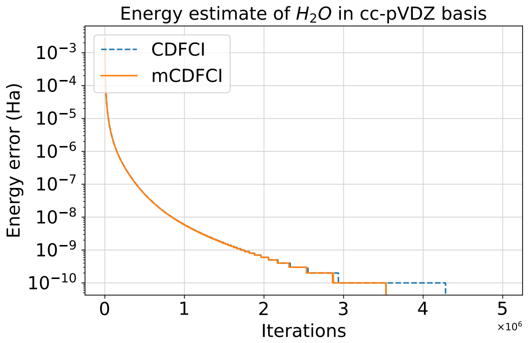
Implementation Details

- H_{ij} evaluated on-the-fly.
- $\mathbf{c}$ and $\mathbf{b} = H\mathbf{c}$ are stored in a hash table.
- Update of $\mathbf{b}_i$ is discarded if $\mathbf{c}_i = 0$ and $\mathbf{\Delta b}_i < \tau$, where $\mathbf{\Delta b} = H_{:,j} a_j$. Note that this does not affect eigenvalue estimator

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

- Compression tolerance τ balances between memory-cost and accuracy.
- Shared memory parallelism based on OpenMP: the updates of $\mathbf{c}$ and $\mathbf{b} = H\mathbf{c}$ for each coordinate are performed in parallel.

Overall Speedup: H_2O/cc-pVDZ



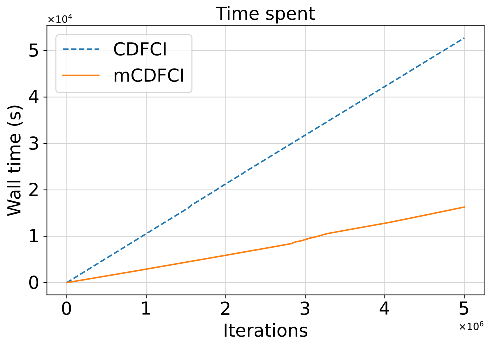
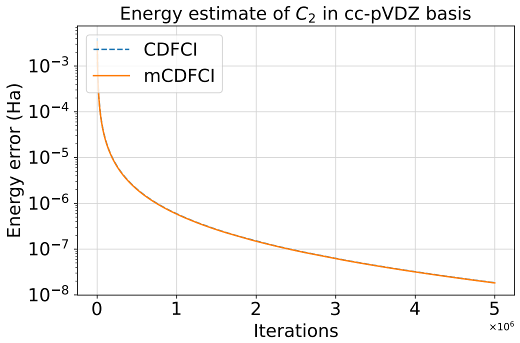


Figure: Speedup of mCDFCI compared with CDFCI(2019), both in 64 threads. We perform k coordinates ($k = 64$) descent per iteration for the original CDFCI.

Overall Speedup: C_2/cc-pVDZ



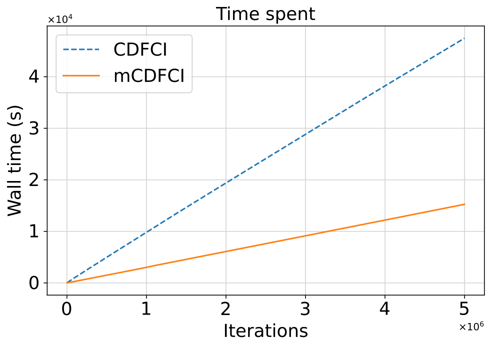
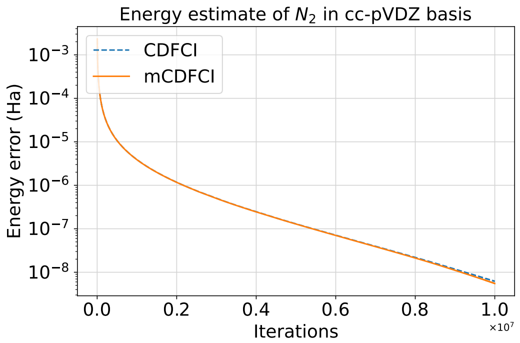
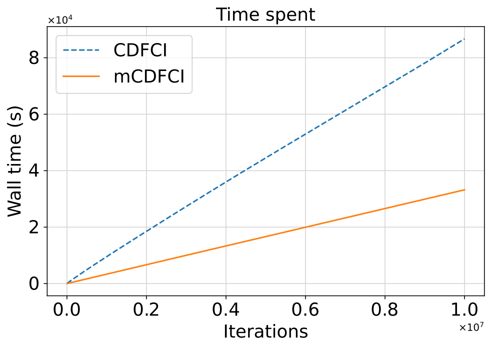


Figure: Speedup of mCDFCI compared with CDFCI(2019), both in 64 threads. We perform k coordinates ($k = 64$) descent per iteration for the original CDFCI.

Overall Speedup: N_2/cc-pVDZ





Remark: With a larger search space each step, mCDFCI leads to lower energy with just 0.2% more coordinates searched.

Scalability

Table: Speedup of mCDFCI for different number of coordinates (threads) of H_2O in cc-pVDZ basis

		Wall Time (s)	Speedup on k Cores					
Energy	Error	on Single Core	2	4	8	16	32	64
-76.2318601	10^{-2}	9.0	1.9×	3.7×	5.0×	7.0×	10.4×	13.1×
-76.2408601	10^{-3}	292.9	2.0×	3.7×	4.8×	8.2×	14.3×	20.3×
-76.2417601	10^{-4}	1837.6	2.0×	3.5×	6.1×	8.3×	16.2×	22.5×
-76.2418501	10^{-5}	9016.7	2.1×	4.1×	8.0×	12.3×	21.1×	29.0×
-76.2418591	10^{-6}	32931.7	2.1×	4.5×	9.3×	16.2×	29.1×	40.7×

Scalability for each procedure

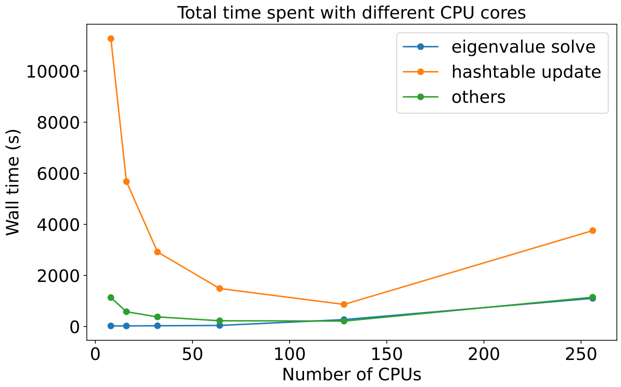


Figure: Time spent listed in each procedure, while running 6.4M core $\times$ iterations for Cr_2 in Ahlrics SV Basis $(48e, 84o)$, $\tau = 10^{-4}$.

Summary

The proposed methods **CDFCI** and **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.

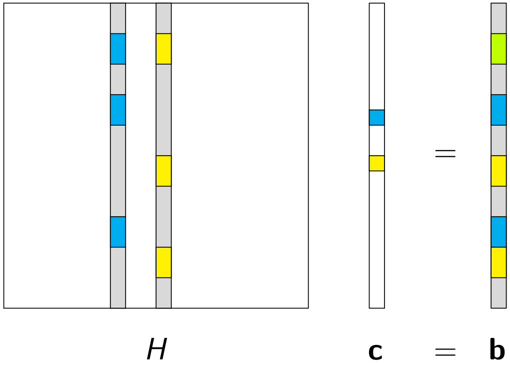


Figure: Update for two coordinates.

Thanks for Your Attention!