B: Exact Diagonalization

B.1: Binary Representation of QMB states and operators

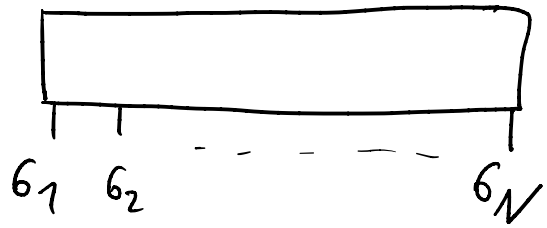
- Start by considering a $S = \frac{1}{2}$ lattice system with N sites.

- Hilbert space $\mathcal{H} = h^{(1)} \otimes \cdots \otimes h^{(N)}$, where $h^{(i)}$ is a two-dimensional complex Hilbert space for site j with local basis $\{ |\uparrow\rangle^{(j)}, |\downarrow\rangle^{(j)} \}$

- QMB state $|\psi\rangle = \sum_{\{\sigma_j = \uparrow,\downarrow\}_j} \psi(\sigma_1, \ldots, \sigma_N) |\sigma_1, \sigma_2, \ldots, \sigma_N\rangle$

$|\sigma_1, \sigma_2, \ldots, \sigma_N\rangle := |\sigma_1\rangle \otimes \cdots \otimes |\sigma_N\rangle$

→ Visualization: $\Psi(\sigma_1, \dots, \sigma_N) \simeq$



- Basic idea: Map basis states $|\sigma_1, \dots, \sigma_N\rangle$ to N-bit integers by identifying $\uparrow \simeq 1$, $\downarrow \simeq 0$, and defining

$$I_\sigma := \sum_{j=1}^{N} \sigma_j 2^{N-j}$$

(σ: shorthand for $(\sigma_1, \dots, \sigma_N)$)

⇝ Spin configurations such as $|\uparrow\uparrow\downarrow\uparrow \dots \downarrow\rangle \simeq |1101 \dots 0\rangle$ are simply read as binary representations of the associated integers I_σ.

⇝ Efficient labelling of full Hilbert space basis coined computational basis

- Generic QMB operators are, due to their linearity, fully specified by their action on the computational basis states.

- Basic examples of operators:

  ↳ $S_z^{(j)} := \mathbb{1} \otimes \cdots \otimes S_z \otimes \cdots \otimes \mathbb{1}$, with $S_z = \frac{\sigma_z}{2}$, is

  (S_z at pos. j)

  diagonal in computational basis:

$$S_z^{(j)} |\ldots, \sigma_j, \ldots\rangle = \underbrace{\frac{2\sigma_j - 1}{2}}_{= m_{\sigma_j} = \pm \frac{1}{2}} |\ldots, \sigma_j, \ldots\rangle$$

↳ Spin-flip: $S_x^{(j)} |\ldots, \sigma_j, \ldots\rangle = \frac{1}{2} |\ldots, 1-\sigma_j, \ldots\rangle$.

↳ Flip-flop: $S_+^{(i)} S_-^{(j)} |\ldots, \sigma_i, \ldots, \sigma_j, \ldots\rangle =$
$(i \neq j)$

$$= \sigma_j (1-\sigma_i) |\ldots, 1-\sigma_i, \ldots, 1-\sigma_j, \ldots\rangle$$

[with $S_\pm = S_x \pm i S_y = \frac{1}{2}(\sigma_x \pm i\sigma_y) = \left\{ \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, + \;;\; \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, - \right\}$]

- Action of QMB operators can be conveniently implemented with bitwise operations.

Recap bitwise operations:

- Apply logical operations such as AND, OR, XOR, bit by bit to a pair of integers in binary representation and store the result in a new integer.
- Defining examples

True $\simeq$ 1
False $\simeq$ 0

int x	int y	x AND y	x OR y	x XOR y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

- For now, think of operators as $2^N \times 2^N$ matrices

$$O = \sum_{I,J} \langle J|O|I\rangle \; |J\rangle\langle I|$$

(later: sparse matrices, on the fly algorithms, symmetries, etc.)

- Examples:

↳ Implementation of $S_z^{(j)}$ from its action on basis integers I.

int jbit = 2^{N-j} ; initialize $S_z^{(j)}$ as zero matrix ;

Loop on $I: 0, \cdots, 2^{N-1}$ {

if ((jbit AND I) $\neq 0$) {

$S_z^{(j)}[I,I]$ += 0.5 ;

} else { $S_z^{(j)}[I,I] -= 0.5$; }

}

↳ Implementation of $S_x^{(j)}$:

⋮

$S_x^{(j)}[I \text{ XOR } jbit, I] += 0.5$;

⋮

↳ Implementation of $S_+^{(i)} S_-^{(j)} + h.c. \quad (i \neq j)$

int bitij = $2^{N-i} + 2^{N-j}$ [% mask for selecting i and j]

Loop on I ⋮

int I_{ij} = I AND bit_{ij}; [% retrieve bits i and j from I]

int hop_{ij} = bit_{ij} XOR I_{ij}; [% Flip bits i and j in I_{ij}]

if $\left((hop_{ij} \neq 0) \wedge (hop_{ij} \neq bit_{ij}) \right) \{ \left(S_+^{(i)} S_-^{(j)} + h.c. \right) [I - I_{ij} + hop_{ij}, I] \; t = 1.0; \}$

[false if both bits i and j are 1 in I]

[logical "and" (not bitwise)]

[false if both bits were 0 in I]

[replace bits i,j by flipped bits in target state]

⋮

— General approach: Decompose generic operator

$$O = \sum_{I,J} \langle J|O|I\rangle \, |J\rangle\langle I|$$ (e.g. the Hamiltonian) into a sum $O = \sum_{r=1}^{F} O_r$ such that for a given I_0, the matrix element $\langle J|O|I_0\rangle$ is non-zero for at most one state J.

⇝ All finite matrix elements are known after a single loop on I, rather than a costly double loop on I and J.

$S_+^{(i)} S_-^{(\vec{\sigma})} + \text{h.c.}$, $S_z^{(j)}$, $S_x^{(\vec{\sigma})}$ are all eligible O_λ's

⇝ Finite elements $\langle J|O|I\rangle$ are stored more economically in a sparse matrix.

↳ # of non-zero elements $\leq F \cdot \dim(\mathcal{H}) = F \cdot 2^N \sim N 2^N$.
(spin $\frac{1}{2}$; typically)

↳ Typically in ED for QMB systems, $F \sim \log(\dim \mathcal{H}) \sim N$, even for spatially non-local Hamiltonians.