Spin-1/2 Chain: Hamiltonians and Observables

Nicolas Rojo

1 Objective of the Code

The goal of this program is to build the main operators required to model a spin-$\frac{1}{2}$ chain from a set of input parameters. The input consists of:

- The number of spins N (`nsites`),
- A site index k where an external field is applied (`sitefield`),
- A target chain state (`op_state_in`) defining the observable we want to maximize.

From these inputs, the code constructs three operators (stored as matrices):

- **Free Hamiltonian $\mathbf{H}_{\text{ff}}$** (`Hff`): computed from N, representing the intrinsic spin–spin interactions of the chain.
- **Interaction / field term $\mathbf{H}_{\text{int}}$** (`Hint`): computed from `sitefield`, representing the action of an external transverse field on a single spin in the chain.
- **Observable operator O** (`opobs`): computed from `op_state_in`. This is an operator associated with a chosen target state of the chain.

In order to optimize memory usage, all operators are internally stored keeping only the diagonal and the upper triangular part of each matrix. This representation is sufficient due to the known Hermitian structure of the Hamiltonians and the observable operator.

The constructed operators are intended to be used as inputs for a quantum control algorithm based on Krotov's method.

2 Construction of the Free Hamiltonian $\mathbf{H}_{\text{ff}}$

Hilbert space, basis, and binary ordering

For a chain of N spin-$\frac{1}{2}$ particles, the Hilbert space dimension is

$$\dim \mathcal{H} = 2^N.$$

The code adopts the computational basis ordered by increasing binary index. Each basis state is labelled by an integer

$$\mathrm{id} \in \{0, 1, \ldots, 2^N - 1\},$$

and is interpreted as a bitstring $b_N b_{N-1} \ldots b_1$ with $b_j \in \{0, 1\}$. The corresponding matrix row index is

$$\mathrm{row} = \mathrm{id} + 1.$$

Example ($N = 4$). The first rows correspond to:

row	$b_4 b_3 b_2 b_1$
1	0000
2	0001
3	0010
4	0011
5	0100
⋮	⋮

Operator form implemented in the code

The free Hamiltonian is built as a nearest-neighbour spin–spin coupling written in the $\{S^z, S^\pm\}$ form:

$$\mathbf{H}_{\mathrm{ff}} = -\sum_{i=1}^{N-1} \left(S_i^z S_{i+1}^z + \frac{1}{2} \left(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+ \right) \right).$$

This structure produces:

- Diagonal terms from $S_i^z S_{i+1}^z$,
- Off-diagonal terms that swap neighbouring $01 \leftrightarrow 10$ configurations.

How matrix elements are generated

For each basis state $|\mathrm{id}\rangle \equiv |b_N \ldots b_1\rangle$ the code loops over all pairs of spins $(i, i+1)$ and extracts the two neighbouring bits (b_i, b_{i+1}).

Diagonal contribution. Using the eigenvalues $S^z |\uparrow\rangle = \frac{1}{2} |\uparrow\rangle$ and $S^z |\downarrow\rangle = -\frac{1}{2} |\downarrow\rangle$, each pair of spins contributes

$$\left\langle S_i^z S_{i+1}^z \right\rangle = \begin{cases} +\frac{1}{4}, & b_i = b_{i+1}, \\ -\frac{1}{4}, & b_i \neq b_{i+1}. \end{cases}$$

The code accumulates a value `diagval` equal to the sum of these $\pm\frac{1}{4}$ contributions, and then stores the diagonal matrix element as

$$(\mathbf{H}_{\mathrm{ff}})_{\mathrm{row,row}} = -\,\texttt{diagval}.$$

Off-diagonal contribution. Whenever a neighbouring pair is different ($b_i \neq b_{i+1}$), the term

$$-\frac{1}{2}\left(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+\right)$$

connects $|\ldots 01\ldots\rangle$ with $|\ldots 10\ldots\rangle$. In practice, the code constructs a new state $\left|\mathrm{id}'\right\rangle$ by swapping the two bits at sites i and $i+1$, and inserts the off-diagonal element

$$(\mathbf{H}_{\mathrm{ff}})_{\mathrm{row,col}} = -\frac{1}{2}, \quad \mathrm{col} = \mathrm{id}' + 1.$$

Worked examples ($N = 4$)

Below, states are written as $b_4 b_3 b_2 b_1$ (with b_1 the rightmost bit).

Example 1: $|0000\rangle$. All three pairs of spins are equal (00), so

$$\texttt{diagval} = 3 \times \frac{1}{4} = \frac{3}{4} \quad \Longrightarrow \quad \mathbf{H}_{\mathrm{ff}} |0000\rangle = -\frac{3}{4} |0000\rangle .$$

There are no 01/10 pairs, so no off-diagonal terms appear.

Example 2: $|0001\rangle$. The pairs are:

$$(4,3) : 00 \ \ \text{(different)}, \qquad (3,2) : 00 \ \ \text{(equal)}, \qquad (2,1) : 01 \ \ \text{(equal)}.$$

Hence

$$\texttt{diagval} = -\frac{1}{4} + \frac{1}{4} + \frac{1}{4} = \frac{1}{4} \quad \Longrightarrow \quad \text{(diagonal)} = -\frac{1}{4} |0001\rangle .$$

The only flip occurs on bond $(2,1)$, swapping $01 \to 10$, which gives

$$|0001\rangle \ \longrightarrow \ |0010\rangle .$$

Therefore,

$$\mathbf{H}_{\mathrm{ff}} |0001\rangle = -\frac{1}{4} |0001\rangle - \frac{1}{2} |0010\rangle .$$

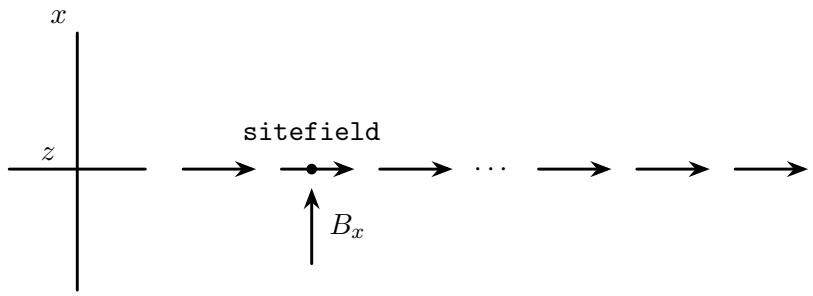
Memory optimization

To reduce memory usage, the code stores only the diagonal and the upper triangular entries (col > row) of $\mathbf{H}_{\mathrm{ff}}$.

3 External-Field Hamiltonian $\mathbf{H}_{\mathrm{int}}$

External transverse field

We consider a chain of N spins initially aligned along the z direction, while an external control field B_x points along the x direction and acts only on a single site k (`sitefield`). A schematic view is shown below.



Operator form

A transverse field along x acting locally on site k is represented by a single spin operator:

$$\mathbf{H}_{\text{int}} = S_k^x$$

Using $S^x = \frac{1}{2}\sigma_x$ with

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

the action of S_k^x in the computational basis is to flip the spin at site k:

$$S_k^x \left| b_N \cdots b_k \cdots b_1 \right\rangle = \frac{1}{2} \left| b_N \cdots (b_k \oplus 1) \cdots b_1 \right\rangle,$$

where $\oplus$ denotes XOR.

How matrix elements are generated in the code

For each basis state labelled by an integer id (row row = id + 1), the code builds

$$\text{id}' = \text{id} \ \oplus \ 2^{N-k}, \qquad \text{col} = \text{id}' + 1,$$

and inserts a single off-diagonal matrix element:

$$(\mathbf{H}_{\text{int}})_{\text{row,col}} = \frac{1}{2}.$$

No diagonal contributions appear for this operator.

Worked example ($N = 4$)

States are written as $b_4 b_3 b_2 b_1$ (with b_1 the rightmost bit), and k counts from the left ($k = 1$ is b_4). Let the field act on site $k = 2$. Consider the basis state

$$|0001\rangle \quad (\text{id} = 1).$$

Flipping bit b_3 maps

$$|0001\rangle \longrightarrow |0101\rangle \quad (\text{id}' = 5),$$

therefore

$$\mathbf{H}_{\text{int}} |0001\rangle = \frac{1}{2} |0101\rangle .$$

Likewise,

$$\mathbf{H}_{\text{int}} |0101\rangle = \frac{1}{2} |0001\rangle ,$$

so the operator couples pairs of states that differ only at site k.

Memory optimization

As with $\mathbf{H}_{\text{ff}}$, the implementation stores only the diagonal and the upper-triangular part (col $>$ row) of $\mathbf{H}_{\text{int}}$.

4 Target State and Projector $\mathbf{O} = |\phi\rangle \langle\phi|$ construction

Input format and meaning of `op_state_in`

The observable is defined from an input string or a list of strings separated by commas of length N composed of the characters

$$\{\texttt{U}, \texttt{R}, \texttt{D}, \texttt{L}\}.$$

Each character specifies the local spin direction at a given site:

- `R` : spin eigenstate along $+z$ ("right"),
- `L` : spin eigenstate along $-z$ ("left"),
- `U` : spin eigenstate along $+x$ ("up"),
- `D` : spin eigenstate along $-x$ ("down").

In the computational basis $\{|0\rangle, |1\rangle\}$ (with $|0\rangle \equiv |+z\rangle$ and $|1\rangle \equiv |-z\rangle$), the local states are:

$$|R\rangle \equiv |+z\rangle \equiv |0\rangle , \tag{1}$$

$$|L\rangle \equiv |-z\rangle \equiv |1\rangle , \tag{2}$$

$$|U\rangle \equiv |+x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|R\rangle + |L\rangle}{\sqrt{2}}, \tag{3}$$

$$|D\rangle \equiv |-x\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|R\rangle - |L\rangle}{\sqrt{2}}. \tag{4}$$

Therefore, any occurrence of `U` or `D` in the input introduces a local superposition in the z-basis.

Target state and observable operator

From the input, the code builds a (normalized) target state

$$|\phi\rangle = \sum_a c_a \, |a\rangle \, ,$$

where $|a\rangle$ are computational basis states (binary-ordered) and the real coefficients c_a arise from expanding all `U`/`D` symbols into $(|0\rangle \pm |1\rangle)/\sqrt{2}$.

The observable is then defined as the projector onto that target state:

$$\mathbf{O} \equiv |\phi\rangle \, \langle\phi| \, .$$

For any system state $|\Psi\rangle$, the expectation value becomes

$$\langle \mathbf{O} \rangle_\Psi = \langle\Psi| \, \mathbf{O} \, |\Psi\rangle = |\, \langle\phi|\Psi\rangle \, |^2,$$

which quantifies the overlap with the desired target configuration.

Multiple input strings

The input `op_state_in` may contain several strings separated by commas, e.g.

$$\texttt{op\_state\_in} = \texttt{''UUUU,DDDD''}.$$

Each string defines a product state $|\phi_m\rangle$ obtained by mapping

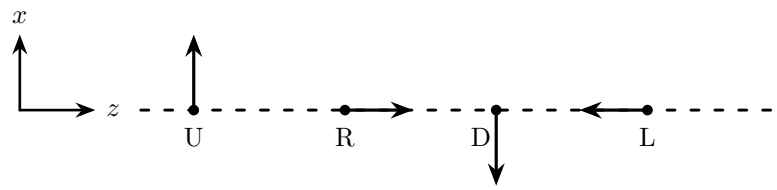
$$\texttt{R} \mapsto |0\rangle \, , \qquad \texttt{L} \mapsto |1\rangle \, , \qquad \texttt{U} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad \texttt{D} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

If M comma-separated strings are provided, the code constructs the target state as an equally-weighted superposition:

$$|\phi\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |\phi_m\rangle \, .$$

Worked example: `URDL` ($N = 4$)

Considering the input string `URDL`.



Interpreting each letter as a single-site state, the target product state is

$$|\phi\rangle = |U\rangle \otimes |R\rangle \otimes |D\rangle \otimes |L\rangle \,.$$

Using $|U\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $|D\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, $|R\rangle = |0\rangle$ and $|L\rangle = |1\rangle$, we obtain:

$$|\phi\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes |0\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \otimes |1\rangle \tag{5}$$

$$= \frac{1}{2}\Big(|0\,0\,0\,1\rangle - |0\,0\,1\,1\rangle + |1\,0\,0\,1\rangle - |1\,0\,1\,1\rangle \Big). \tag{6}$$

Hence the observable matrix is

$$\mathbf{O} = |\phi\rangle \langle\phi| \,,$$

which in the computational basis has elements $O_{ab} = c_a c_b$ for the four basis states that appear in the expansion above, and zero in the rest.

Memory optimization

As with the Hamiltonians, the observable operator $\mathbf{O}$ is stored internally by keeping only the diagonal and the upper-triangular entries to minimize memory usage.

5 Output files (`print_matrix` option)

The code includes an optional output mode controlled by the logical flag `print_matrix`. When `print_matrix = .true.`, the program reconstructs the full form of each operator and writes the resulting matrices into text files:

- `Hff.txt`: the free Hamiltonian $\mathbf{H}_{\text{ff}}$,
- `Hint.txt`: the local field (interaction) Hamiltonian $\mathbf{H}_{\text{int}}$,
- `Opobs.txt`: the observable operator $\mathbf{O} = |\phi\rangle \langle\phi|$.

These files can be used for inspection or debugging.