Exercise 6.1: Exact diagonalization with quantum numbers

Download the script `ed_conserve.py` from the website. It implements the same Hamiltonian as in the last weeks, namely

$$H = -J \sum_{j=0}^{L-1} \sigma_j^x \sigma_{j+1}^x - g \sum_{j=0}^{L-1} \sigma_j^z, \tag{1}$$

a) Call the function `calc_H()` for $N = 10, J = 1, g = 0.1$ to obtain a dictionary of block-Hamiltonians. Determine the ground state energy using `scipy.sparse.linalg.eigsh` and ensure that you get the same result as in last weeks program (which should be $E_0 \approx -10.0250156642343$).

b) We identify spin configurations with integers using the binary representation, e.g. for 6 sites,

$$|\downarrow\uparrow\downarrow\uparrow\uparrow\downarrow\rangle = 010110_2 = 16 + 4 + 2 = 22_{10} \tag{2}$$

The builtin Python functions `bin()` and `int()` are useful to convert between these representations and are used in the function `ed_conserve.translate()` to shift the bits, implementing the translation operator T. However, this implementation is fairly slow (and actually a performance critical part of the program). Since the computer stores integers in binary form anyways, it is naturally to directly use the bitwise operators `&` (AND), `|` (OR), `^` (XOR), and `>>`, `<<` (for right, left shift of the bits). Replace the implementation of the `translate()` function by a faster version using only the bitwise operators.

c) One advantage of the block diagonal form is that we can directly label the energies by k and e.g. inspect the dispersion relation of excitations. Plot the lowest 5 energies in each k block versus the momentum quantum number k for $N = 14, g = 1$.

d) Call the function `ed_conserve.calc_basis()` and extract the dimensions of the blocks. Plot the dimensions of the blocks versus N on a logarithmic y-scale.

e) While the code uses momentum conservation, it does not exploit the parity symmetry: the operator $P = \prod_{j=0}^{N-1} \sigma^z$ with eigenvalues $p = \pm 1$ commutes with both H and T. Adjust the functions `ed_conserve.calc_basis()` and `ed_conserve.calc_H()` such that they exploit P for a further block-diagonalization of H.

Hint: Write a function to determine the parity eigenvalue p for a given spin configuration. Use tuples `(p, k)` (instead of simply `k`) as keys `qn` for the dictionaries `basis,ind_in_basis, H` and adjust code using these keys. That's all!

f) Include the blocksizes when using parity (with a different color) into the plot of part d).

g) Regenerate the plot of c) but use two different colors for $p = \pm 1$. Generate a plot each for combination of $J \in \{+1, -1\}$ and $g \in \{0.5, 1, 1.5\}$. What do you observe?

Exercise 6.2: Area law - ground state versus random state

We consider a chain of even length L of spin-$\frac{1}{2}$ degrees of freedom; the Hilbert space has thus dimension 2^L. Moreover, we consider half-chain bipartitions into regions A (left half) and B (right half).

a) Using exact diagonalization from the exercise sheet of the last weeks, generate the ground state of the transverse field Ising model for $L = 14, g = 1.5, J \equiv 1$ for **open** boundary conditions. The state should be given by 2^L complex numbers ψ_i (arranged into a 1D array) such that

$$|\psi\rangle = \sum_{i=0}^{2^L-1} \psi_i |i\rangle = \sum_{j_1,\ldots,j_L \in \{0,1\}} \psi_{j_1,\ldots,j_L} |j_1,\ldots,j_L\rangle , \tag{3}$$

where we identify the basis states with binary representations of the index i (see also the next exercise), e.g., for $L = 6$ we have

$$|\uparrow\downarrow\uparrow\downarrow\downarrow\uparrow\rangle \equiv |j_1 = 0, j_2 = 1, j_3 = 0, j_4 = 1, j_5 = 1, j_6 = 0\rangle = |i = 010110_2 = 22_{10}\rangle .$$

b) Convince yourself that `psi_ab = np.reshape(psi_i, (2**(L//2), 2**(L//2)))` arranges the state into the form

$$|\psi\rangle = \sum_{a=0}^{2^{\frac{L}{2}}-1} \sum_{b=0}^{2^{\frac{L}{2}}-1} \psi_{a,b} |a\rangle_A |b\rangle_B , \tag{4}$$

where $|a\rangle_A$ labels the basis states in the left half A, and $|b\rangle_B$ in the right half, respectively. (The result $\psi_{a,b}$ is a 2D array, i.e., a matrix with the same number 2^L of entries as in the original state.)

Hint: In general, $i = a \cdot 2^{\frac{L}{2}} + b$, e.g, for the above example,

$$|\uparrow\downarrow\uparrow\downarrow\downarrow\uparrow\rangle = |i = 010110_2 = 010_2 \cdot \underbrace{1000_2}_{=8_{10}=2^3} + 110_2\rangle = |a = 010_2\rangle_A |b = 110_2\rangle_B . \tag{5}$$

c) Find the Schmidt decomposition $|\psi\rangle = \sum_\alpha \lambda_\alpha |\alpha\rangle_A |\alpha\rangle_B$ by performing a singular value decomposition of $\psi_{a,b}$.

Hint: Here and in the following: if you get a `LinAlgError: SVD did not converge`, try using `scipy.linalg.svd(...., lapack_driver='gesvd')`.

d) Sort the Schmidt values descending and plot the largest 20 Schmidt values λ_α versus the index α. Use a logarithmic y-axis for the Schmidt values λ_α.

e) Generate a random state drawn from the Haar measure of the full Hilbert space. Calculate its Schmidt spectrum $\tilde{\lambda}_\alpha$ as in c) and include it into the plot of d).

Hint: To draw a state from the Haar measure means to choose a normalized vector (uniformly) pointing in a random direction. You get such a vector if you draw the real and imaginary part of each coefficient from independent normal (gaussian) distributions and finally normalize it.

f) Write a function which calculates the entanglement entropy $S = -\sum_\alpha \lambda_\alpha^2 \log(\lambda_\alpha^2)$ (when given λ_α).

g) Calculate the entanglement entropy S of

- the ground state in the paramagnetic phase ($g = 1.5J$),
- the ground state at the critical point ($g = J$),
- the ground state in the ferromagnetic phase ($g = 0.5J$), and
- a random states drawn from the Haar measure, compare this entropy to the Page value $S_{Page} = \frac{L}{2}\log(2) - \frac{1}{2}$.

for various system sizes $L = 6, 8, 10, 12, 14$. Plot S versus L. What do you observe?