Exercise 5: Exact diagonalization with quantum numbers

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

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

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 first 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?