

Generalities and advice for the B.Sc. project of Tobias

by Yaroslav Herasymenko

We aim to use perturbation theory (PT) series as a prescription for efficient variational quantum ansatzes (VQAs). As shown in our 2019 paper, for the above goal it is useful to split PT series into terms represented by ‘coupling activation diagrams’. One may then build VQAs by mapping those diagrams onto the unitary gates of the VQA. The efficiency of the resulting VQA is ensured by focusing on the diagrams which dominate the PT series. In this project, we will try to automate this whole procedure in a Python module.

The timeline

Here’s the approximate timeline for the project, starting 1st of March.

- 01/03-26/04 : getting the algorithm which (a) generates diagrams and (b) produces the VQAs.
- 15/04-15/05 : some basic physical applications for the algorithm, maybe some extras.
- 15/05-15/06 : writing the thesis.

The details of the algorithm

PT diagram generator

Here you can see some tips and explanations for the diagram generating part of the algorithm. First, let us briefly define here what do we mean by the ‘coupling activation diagrams’. Consider splitting the target Hamiltonian as $H = -\sum h_i Z_i + \sum_{\alpha=1}^{N_c} J_\alpha V_\alpha$, such that V_α are Pauli strings ($J_\alpha \ll h_i$). Then the PT series for the ground state $|\Psi\rangle$ can be organised as a Taylor series:

$$|\Psi\rangle(J_1, \dots, J_{N_c}) = \sum_{k_1, \dots, k_{N_c}=0}^{\infty} J_1^{k_1} J_2^{k_2} \dots J_{N_c}^{k_{N_c}} |\Psi\rangle_{(k_1, \dots, k_{N_c})}, \quad (1)$$

or, in an equivalent but more compact ‘vector power’ notation:

$$|\Psi\rangle(\vec{J}) = \sum_{\vec{k}} \vec{J}^{\vec{k}} |\Psi\rangle_{\vec{k}}. \quad (2)$$

Combining the terms with fixed Manhattan length $|\vec{k}| \equiv k_1 + k_2 + \dots + k_{N_c}$, we get back to the standard PT with $|\vec{k}|$ being the PT order. The series in (2), however, is much more informative. The key property of separate terms $|\Psi\rangle_{\vec{k}}$ is that of being proportional to a computational basis state of a very specific form:

$$|\Psi\rangle_{\vec{k}} = C_{\vec{k}} \vec{V}^{\vec{k}} |\vec{0}\rangle, \quad (3)$$

where $C_{\vec{k}}$ is just some real-valued coefficient. This allows to think of \vec{k} as certain ‘activation patterns’ of the coupling terms $J_\alpha V_\alpha$. Specifically, pattern $\vec{k} = (k_1, \dots, k_{N_c})$ implies that the coupling $J_1 V_1$ is ‘activated’ k_1 times, $J_2 V_2$ – k_2 times, etc. This pattern can be neatly depicted as a diagram. The diagrams can be classified as connected or disconnected. See the paper for more detailed explanations and illustrations.

Our goal is, given the Hamiltonian, to generate all leading order connected \vec{k} up to certain $k = |\vec{k}|$. The ‘leading order’ property means that we ignore the patterns \vec{k} which produce same states $\vec{V}^{\vec{k}} |\vec{0}\rangle$ while having a higher value of $|\vec{k}|$. The generated activation patterns \vec{k} can then be used in constructing a VQA (see next subsection).

Tips for generating the desired patterns:

1. Build them up incrementally, starting from $\vec{k} = \vec{0}$. First exhaust all \vec{k} for $|\vec{k}| = 1$, then for $|\vec{k}| = 2$, etc.
2. We’ll only need to output the leading order connected contributions. **This allows to save resource also on the intermediate calculations.** Specifically, to find leading order connected contributions \vec{k} , you can ignore all disconnected contributions in the intermediate calculations (check if you agree!). However, you generally can’t ignore the subleading connected contributions (do you agree? can you give an example of such a case?).
3. Most useful information about the contribution is encoded in \vec{k} . So, there’s no need to store and carry around the attributes of the pattern. Instead, realize them as separate functions of \vec{k} . This includes:

- (a) The computational basis state produced by the pattern, i.e. $\vec{V} \cdot \vec{k} |\vec{0}\rangle \equiv \pm i^a |\vec{s}\rangle$ (input - \vec{k} , output - $a \in \{0, 1\}$, \vec{s})
- (b) The diagrammatic representation (input - \vec{k} , output - a graph)
- (c) The connectedness of the contribution (input - \vec{k} , output - $\text{Con}(\vec{k}) = \text{TRUE}/\text{FALSE}$); best is to find connectedness without building the graph of the diagram.

Gate-diagram correspondence

Our VQA construction scheme includes a way of translating the PT diagrams into the VQA gates. We consider size-extensive schemes. There, one needs to introduce new gates only when representing leading order connected diagrams. Examples include: QCA, UCC, Trotterized-UCC. To be specific, let's focus on QCA. Others are important extras and should be coded, too – if we have time.

To Tobias: more specifics will be added when you get here

The full VQA construction algorithm

Input: A Hamiltonian H for an interacting qubit system, whose ground state is to be produced by our variational algorithm. The number of parametrized gates N_p , which we allow for the VQA.

Output: The VQA, represented as a list of gates $U_l(\theta_l)$ applied to the starting state: $\psi(\theta_l) = \prod_l U_l(\theta_l) |\vec{0}\rangle$. Elements of the algorithm:

1. Consider k -th order PT for the ground state of H . Generate all connected PT theory diagrams in that order. (start with $k = 1$)
2. Reproducing some of these new PT theory diagrams with a VQA may require including the corresponding gates $U_l(\theta_l)$. Include these new gates into the VQA. (initially the ansatz contains no gates)
3. Repeat 1-2 for the higher PT order $k = k + 1$, until the VQA has the desired total of N_p gates.

Applications

Extras