Lie Algebraic Approach to Simulation of Spin Systems via Quantum Computers

NC STATE

U.S. DEPARTMENT OF ENERGY

Efekan Kökcü[1], Thomas Steckmann[1], Alexander F. Kemper[1], Eugene F. Dumitrescu[3], Yan Wang[3] Daan Camps[2], Lindsay Bassman[2], Wibe A. de Jong[2], Roel Van Beeumen[2], J.K.Freericks[4]

[1]North Carolina State University, [2]Lawrence Berkeley National Laboratory, [3]Oak Ridge National Laboratory, [4]Georgetown University

Kemper Lab

- Unitary synthesis is the main problem of quantum computation
- Trotterization is the main approach used to generate an approximate circuit. Its depth increases with simulation time t.
- Not reliable with NISQ devices for long simulation times. There are variational [1] and statistical [2] approaches to reduce number of Trotter steps.

- Cartan decomposition was used [3-5] to produce **fixed depth** circuits for any unitary matrix. However due to exponential size of $\mathfrak{su}(2^n)$, number of parameters to fit are exponentially large.
- There are variational approaches to generate **constant depth** circuits [6-7] where [7] specifically takes advantage of the algebra of the systems.

In this poster,

- We provide two methods given in [8] and [9-10]
- First method can generate fixed depth circuits that are shallower than generic unitary ways [3-5]. Moreover, we need only **local minimum** of a cost function as opposed to other variational methods such as [6].
- Second method can compress a given Trotter expansion given a very simple to check set of rules we call **block rules** are satisfied. It is **constructive** in contrast to [7].

Fixed Depth Hamiltonian Simulation via Cartan Decompostion [8]

$$\mathcal{H} = ZZ + b_1 IX + b_2 XI \in \{ZZ, IX, XI\}$$

1 Hamiltonian Algebra

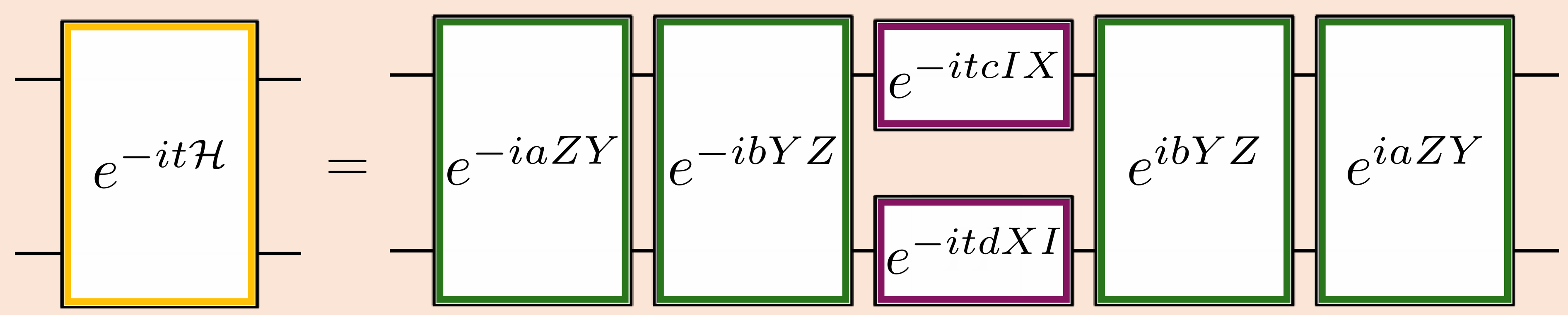
$$\mathfrak{g}(\mathcal{H}) = \begin{Bmatrix} XI,\ IX \\ ZZ,\ YY \\ ZY,\ YZ \end{Bmatrix}$$

$(\mathcal{H} \in \mathfrak{m})$ 2 Cartan Decomposition

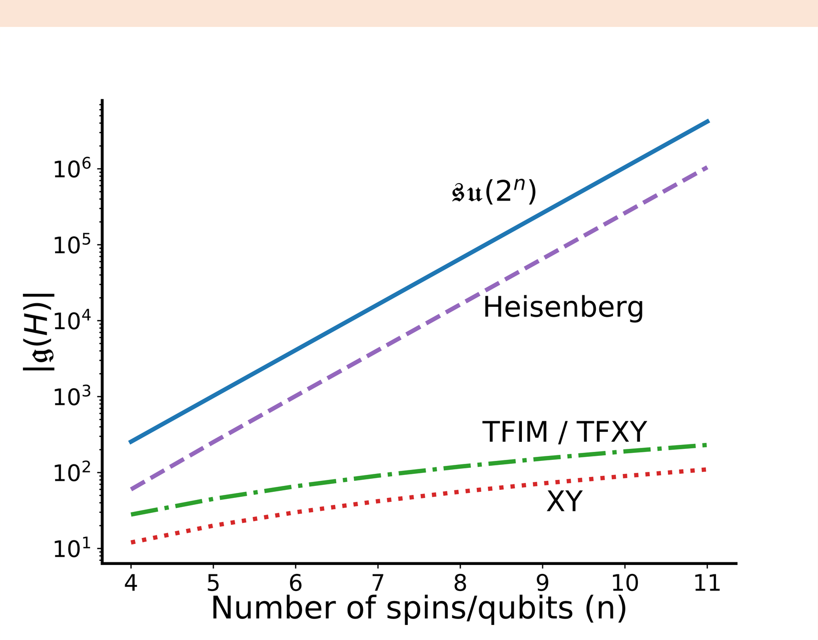
$$\mathfrak{k} = \{ZY,\ YZ\} \qquad \mathfrak{m} = \begin{Bmatrix} ZZ,\ YY \\ XI,\ IX \end{Bmatrix}$$

$\mathfrak{h}$

3 For $K = e^{ibZY}e^{iaYZ} \in e^{\mathfrak{k}}$ and $v = IX + \pi XI \in \mathfrak{h}$, find a **local minimum** of $f(K) = \mathrm{Tr}(KvK^\dagger\mathcal{H})$ which satisfies $K^\dagger\mathcal{H}K \in \mathfrak{h}$. Calculate $K^\dagger\mathcal{H}K \in \mathfrak{h}$, and then build the following time evolution circuit:



Dimension of $\mathfrak{g}(\mathcal{H})$

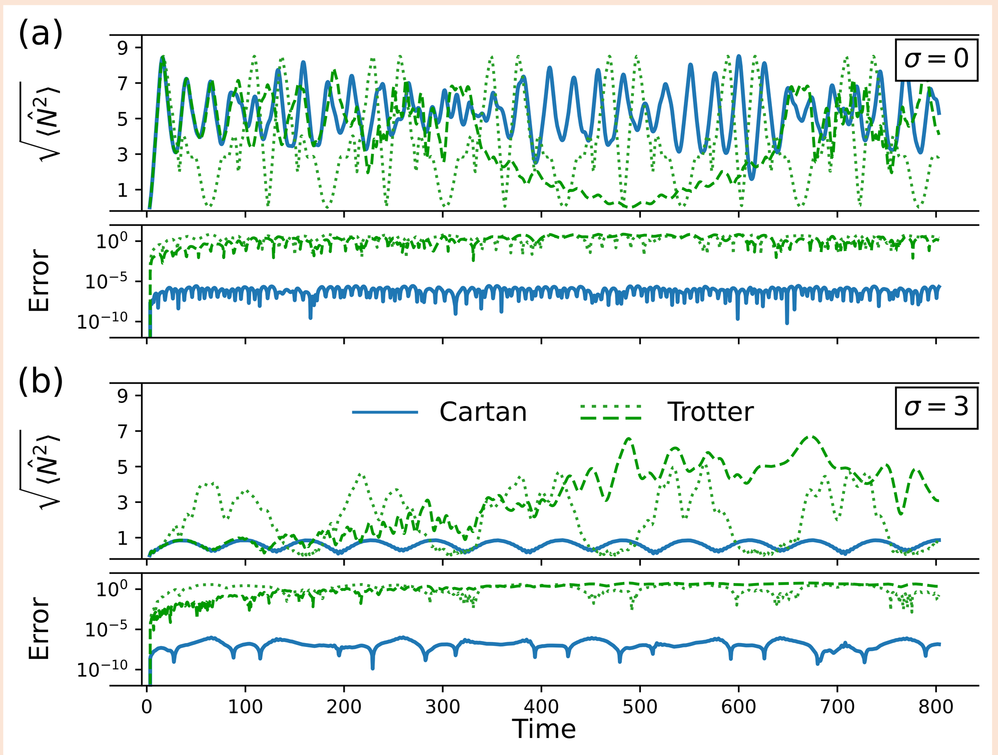


Example: Anderson Localization

- We apply out algorithm on a 10 site Transverse Field XY Hamiltonian with random B field on Z direction with initial state $|\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\rangle$ and measure the location of the spin excitation

$$\hat{N} = \sum_{r=1}^{10}(r-1)\frac{1-Z_r}{2}$$

- With Cartan, simulation error is considerably smaller, and the spin excitation remains trapped as it should due to Anderson Localization [11].

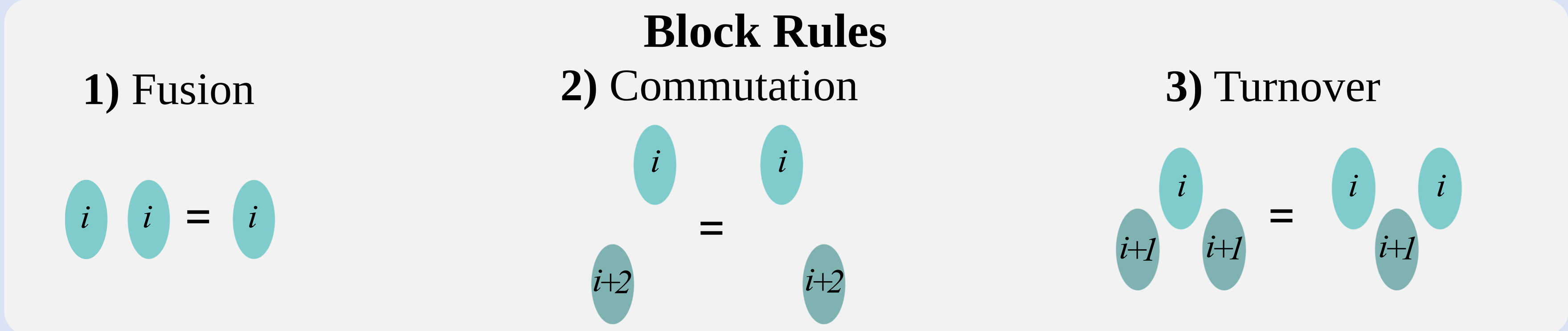


Cartan: 180 Cnots, Trotters: 180 and 1332 Cnots

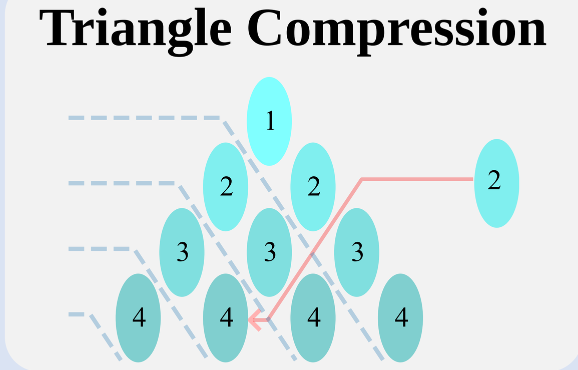
Algebraic Compression of Quantum Circuits [9-10]

Block Rules

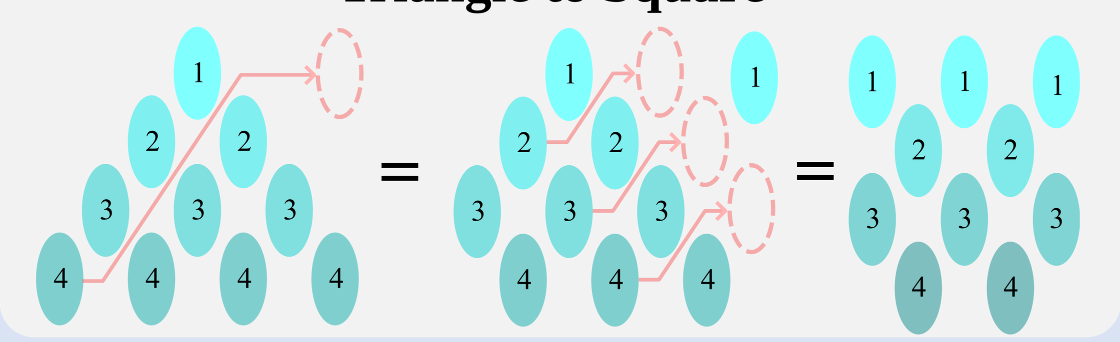
1) Fusion **2)** Commutation **3)** Turnover



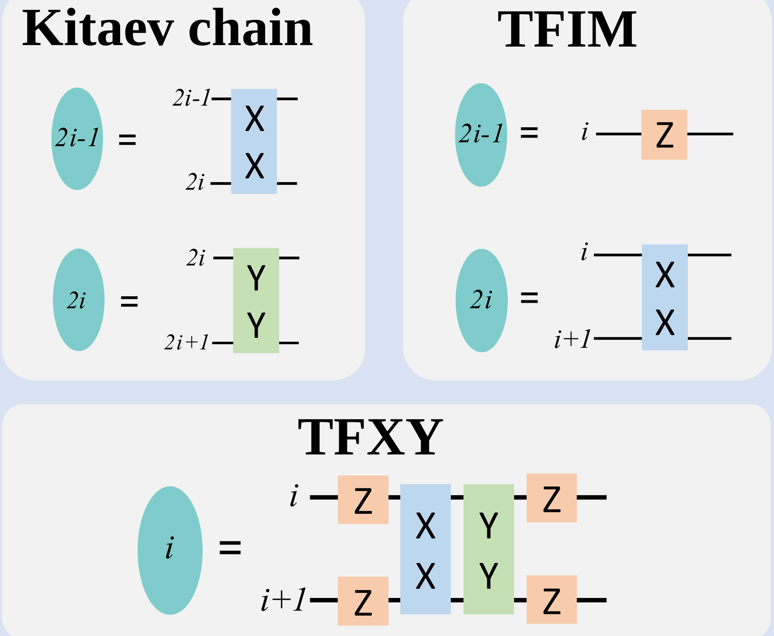
Triangle Compression

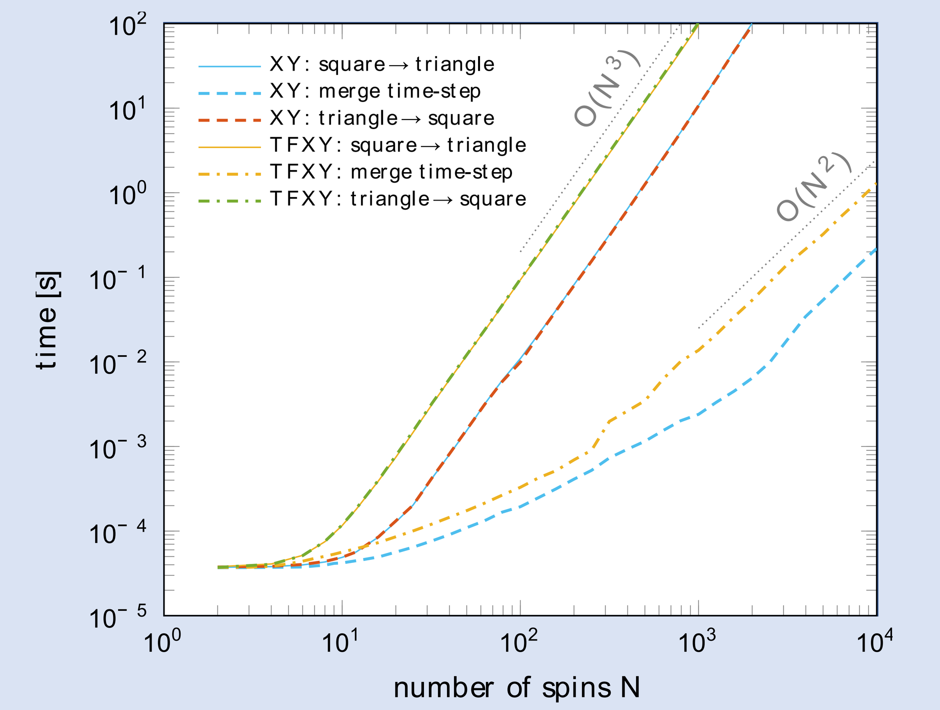


Triangle to Square



Kitaev chain **TFIM** **TFXY**



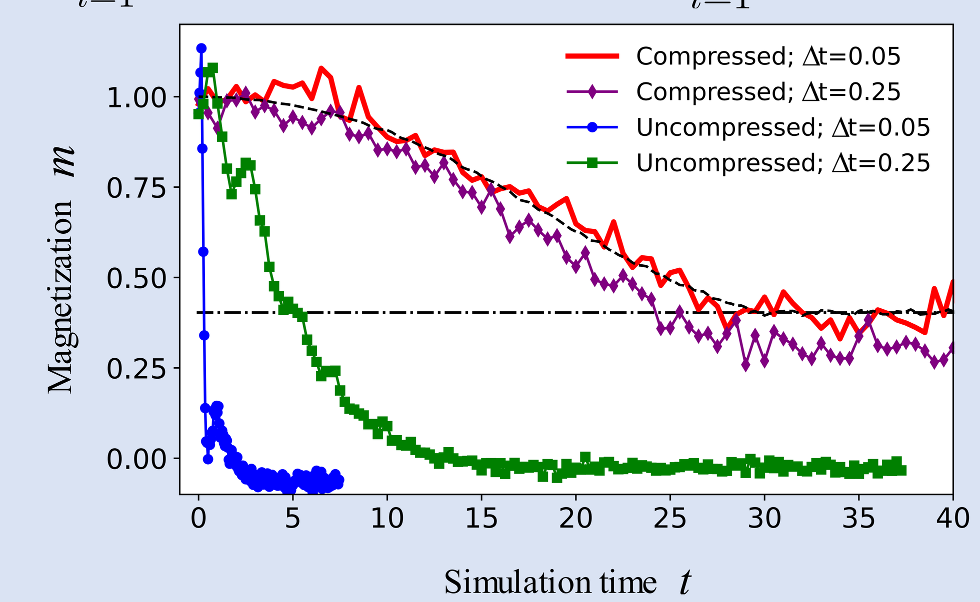


Example: Adiabatic State Preparation

- We apply our algorithm to a 5 site TFIM Hamiltonian with $h_z = -1$ with slowly varying interaction parameter to evolve the ground state of the $J = 0$ system to the ground state of the $J = -2$ system via adiabatic evolution.

$$\mathcal{H}_{ASP}(t) = J(t)\sum_{i=1}^{n-1} X_i X_{i+1} + h_z \sum_{i=1}^{n} Z_i \qquad m = \frac{1}{n}\sum_{i=1}^{n}\langle Z_i\rangle_t$$

- Results from *ibmq_brooklyn.* Trotter circuits contain 8 CNOTs **per time step.** Compressed circuits however contain 20 CNOTs **in total** for all simulation time.



Discussion and Future Work

- We have provided two methods to generate fixed depth circuits, with codes given in [12] and [13].
- Fixed depth circuits are particularly important in NISQ era. Even though it is too deep, as opposed to Trotter, they lead to the similar noise in quantum machine.
- The methods were applied to time evolution but not limited to it. Both methods has a potential to be in a quantum compiler.
- We hope to use Cartan decomposition to generate more efficient approximate circuits rather than exact for more complicated models such as Hubbard model.
- We are currently working on quantum assisted version of the Cartan decomposition method.
- We are working on extending the block structures to generalize it into a broader class of physical models. Be on the lookout for Efekan et.al.

Acknowledgement

We acknowledge the support from US department of Energy with funding numbers DE-SC0019469, DE-AC02-05CH11231, DE-AC05-00OR 22725, ERKJ347; as well as the use of IBM Quantum services.

References

[1] Tran, M. C., Su, Y., Carney, D., & Taylor, J. M. (2021), PRX Quantum, 2(1), 010323.
[2] Zhang, Z. J., Sun, J., Yuan, X., & Yung, M. H. (2020), arXiv:2011.05283.
[3] N. Khaneja and S. J. Glaser, Chemical Physics 267, 11 (2001)
[4] B. Drury and P. Love, Journal of Physics A: Mathematical and Theoretical 41, 395305 (2008).
[5] H. N. S. Earp and J. K. Pachos, Journal of Mathematical Physics 46, 082108 (2005)
[6] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, (2020), npj Quantum Information, 6(1), 1-10.
[7] L. Bassman, R. Van Beeumen, E. Younis, E. Smith, C. Iancu, and W. A. de Jong, arXiv:2103.07429 (2021).
[8] E. Kökcü, T. Steckmann, Y. Wang, J. K. Freericks, E. F. Dumitrescu, and A. F. Kemper, Phys. Rev. Lett. 129, 070501, (2022)
[9] E. Kökcü, D. Camps, L. Bassman, J. K. Freericks, W. A. de Jong, R. Van Beeumen, and A. F. Kemper, Phys. Rev. A 105, 032420 (2022)
[10] D. Camps, E. Kökcü, L. Bassman, W. A. de Jong, A. F. Kemper, and R. Van Beeumen, SIAM J. Matrix Anal. Appl. 43 (2022)
[11] P. W. Anderson, Phys. Rev. 109, 1492 (1958)
[12] E. Kökcü and T. Steckmann, github.com/kemperlab/ cartan-quantum-synthesizer
[13] D. Camps, R. Van Beeumen, E. Kökcü, F3C, F3C++, F3Cpy, https://github.com/QuantumComputingLab.