Tensor Software and Algorithms for Quantum Simulation

Edgar Solomonik

L P N A @CS@Illinois

Department of Computer Science
University of Illinois at Urbana-Champaign

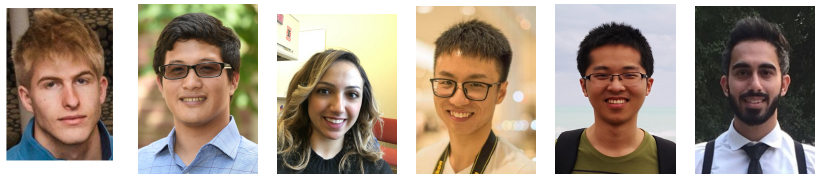
IQUIST Seminar, UIUC

Laboratory for Parallel Numerical Algorithms

Recent/ongoing research topics
(*-covered today)

- parallel matrix computations
  - matrix factorizations
  - eigenvalue problems
  - preconditioners
- tensor computations
  - tensor decomposition*
  - sparse tensor kernels
  - tensor completion
- simulation of quantum systems
  - tensor networks*
  - quantum chemistry*
  - quantum circuits*
- fast bilinear algorithms
  - convolution algorithms
  - tensor symmetry*

L P N A @CS@Illinois



DOE CSGF

NSF

http://lpna.cs.illinois.edu

Outline

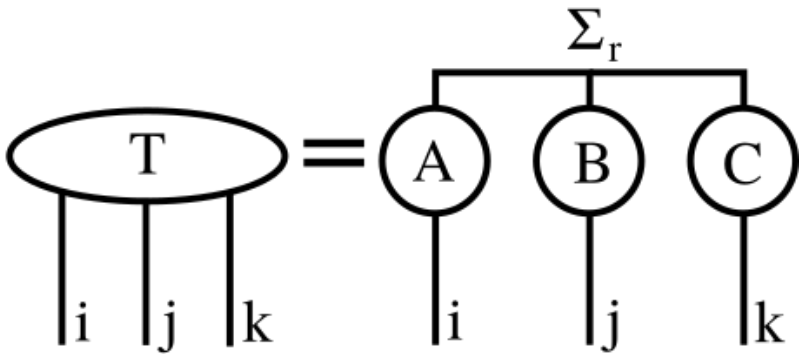
Definitions and Overview

- A tensor of order N has N modes and dimensions $s \times \cdots \times s$
- Two or more tensors can be contracted together in various ways, generalizing matrix/vector products, Hadamard products, etc.
- Tensors decompositions represent a tensor as a contraction of smaller ones (e.g., low-rank matrix factorization)
- Tensor network methods seek to solve eigenvalue/optimization problems with a tensor that is already decomposed
- In the first part of this talk, we look at where tensor contractions and decompositions arise in quantum chemistry methods
- In the second part of this talk, we switch focus to tensor networks and their application both to electronic structure methods and quantum circuit simulation

Tensor Decompositions

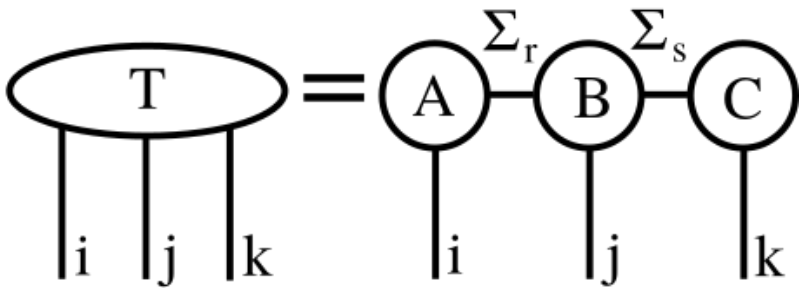
- Canonical polyadic (CP) tensor decomposition[1]

$$t_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$$

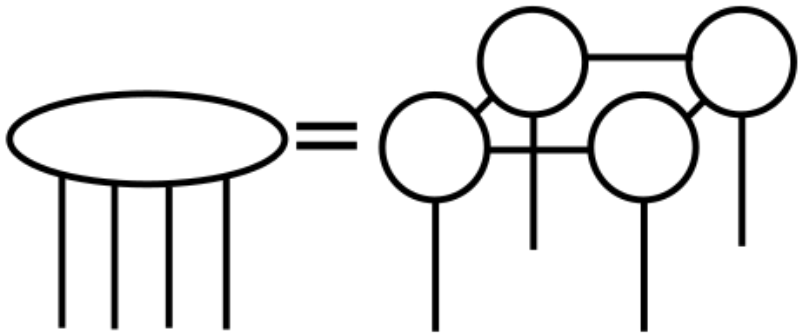


- 1D tensor network / Matrix product state (MPS) / tensor train (TT) decomposition

$$t_{ijk} = \sum_{r} \sum_{s} a_{ir} b_{rjs} c_{sk}$$



- 2D tensor network / projected entangled pair state (PEPS)



[1] T.G. Kolda and B.W. Bader, SIAM Review 2009

Time-Independent Manybody Schrödinger Equation

- To model molecules and solids at a quantum level, we seek low energy configurations in an exponential state space by optimizing over an appropriate subspace
- Given Hamiltonian operator H, seek wavefunction ψ to minimze

$$E = \langle \psi | H | \psi \rangle$$

- H is typically represented as a sum of local operators $H_1, \ldots, H_m$ where $m = O(\text{poly}(n))$

$$H = \sum_{i=1}^{m} H_i$$

  where $H_i|\psi\rangle$ transforms one or two of qubits/particles in ψ
- For simple spin-system models $m = O(n)$, for electronic structure calculations (finding ground state of system of fermions) with a basis set of size $O(n)$, $m = O(n^4)$

Wavefunction Methods in Electronic Structure

- For n-particle systems, the Hamiltonian is described by

$$H = -\frac{1}{2m}\sum_{i=1}^{n}\nabla_i^2 + \sum_{i=1}^{n}V(x_i) + \sum_{i=1}^{n}\sum_{j<i}U(x_i, x_j)$$

- The one-particle component $V(x_i)$ encodes interactions between electrons and atoms
- The two-particle component $U(x_i, x_j)$ encodes electron–electron interactions, specifically $U(x_i, x_j) = -1/|x_i - x_j|$
- Various methods define a subspace by imposing a wavefunction ansatz

$$\psi^{\mathsf{DFT}}(x_1, \ldots, x_n) = \psi_1(x_1)\cdots\psi_n(x_n) \quad \text{(Density Functional Theory)}$$

$$\psi^{\mathsf{HF}}(x_1, \ldots, x_n) = \frac{1}{\sqrt{n!}}\det(\psi_1(x_1), \ldots, \psi_n(x_n)) \quad \text{(Hartree-Fock)}$$

$$\psi^{\mathsf{CCSD}}(x_1, \ldots, x_n) = e^{T_1+T_2}|\psi^{\mathsf{HF}}\rangle \quad \text{(Coupled Cluster)}$$

Electron-Reupsion Integral (ERI) Tensor

- Calculating the expectation value of the two-electron operator Hartree-Fock wavefunction ansatz, we obtain

$$\langle\psi|U|\psi\rangle = \frac{1}{n(n-1)}\sum_{i\neq j}^{n}\langle\psi_i(x_i)\psi_j(x_j)|U(x_i,x_j)|\psi_i(x_i)\psi_j(x_j)\rangle - \langle\psi_i(x_i)\psi_j(x_j)|U(x_i,x_j)|\psi_j(x_i)\psi_i(x_j)\rangle$$

- Given a set of orthogonal basis functions $\chi_1(x),\ldots,\chi_k(x)$, so each single-particle basis function is $\psi_i(x) = \sum_{j=1}^{k} c_{ik}\chi_j(x)$

$$\langle\psi|U|\psi\rangle = \frac{1}{n(n-1)}\sum_{i\neq j}^{n}\sum_{k,l} c_{ik}c_{jl}[(ik|jl) - (il|jk)]$$

where $(ij|kl) = \langle\chi_i^*(x)\chi_j^*(x)|U(x,x')|\chi_k(x')\chi_l(x')\rangle$ is the ERI tensor

- The ERI tensor has *permutational* symmetry $(ij|kl) = (kl|ij) = (kl|ji) = \ldots$ and generally has *group* symmetries due to conservation laws, which permit reduced representations/cost

Hartree-Fock Methods

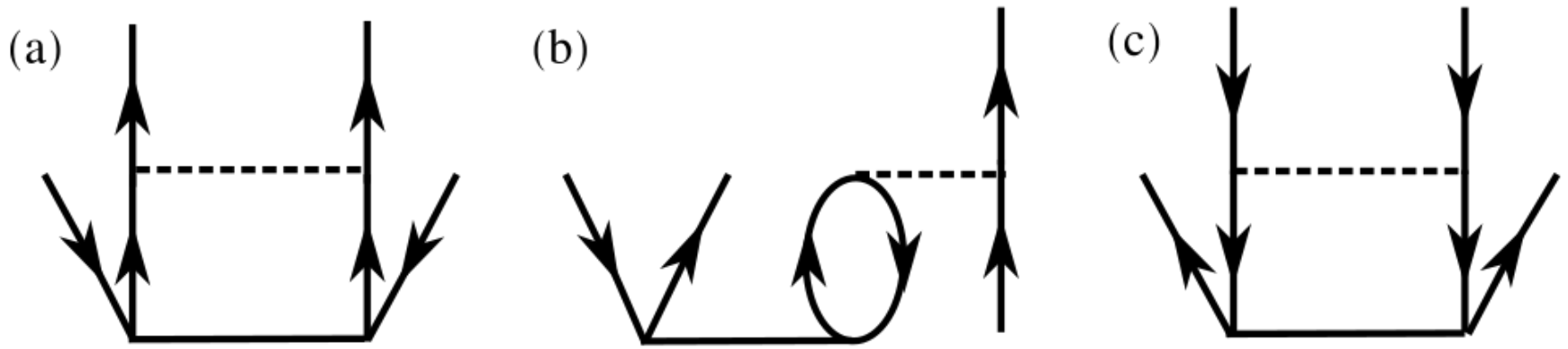
- The Hartree-Fock method computes the best coefficients c_{ik} and obtains ψ^{HF} by iterative minimization via the Self Consistent Field (SCF) procedure
- Hartree-Fock is a mean-field approximation of the potential that takes account electron exchange due to antisymmetrization, but does not model excitations/correlation
- Coupled-cluster methods account for these effects via systematic approximation that also satisfies size extensivity (energy scales correctly with number of non-interacting systems)

Coupled-Cluster Methods

- The singles and double (CCSD) method optimizes amplitude tensors T_1 (order 2) and T_2 (order 4), so as to minimize

$$E \approx \langle \psi^{\text{CCSD}} | H | \psi^{\text{CCSD}} \rangle \quad \text{where} \quad \psi^{\text{CCSD}} = e^{T_1+T_2} | \psi^{\text{HF}} \rangle$$

- Expanding $e^{T_1+T_2}$ and contracting with the two-electron integral tensor, higher-order terms in T_1 and T_2 can be shown to vanish, and the remaining terms are at most as expensive as a contraction of two order 4 tensors into an order 4 tensor, which costs $O(n^6)$



Density Fitting and Tensor Hypercontraction

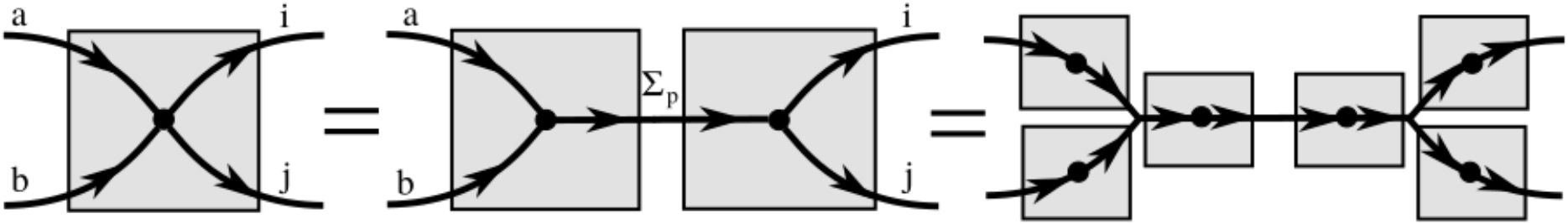
- The cost of CCSD can be reduced to $O(n^5)$ by density fitting, which is a truncated Cholesky decomposition of the ERI tensor

$$(ab|ij) = \sum_p d_{abp} d^*_{ijp}$$

- The tensor hypercontraction (THC) method factorizes the density fitting tensor as

$$d_{ijp} = \sum_r x_{ir} x_{jr} y_{pr}$$

  which is a *canonical polyadic (CP) decomposition* with a repeating factor matrix $\boldsymbol{X}$



CP Decomposition for Tensor Hypercontraction

- The tensor hypercontraction (THC) method factorizes the density fitting tensor as

$$d_{ijp} = \sum_r x_{ir} x_{jr} y_{pr}$$

  which is a *canonical polyadic (CP) decomposition* with a repeating factor matrix $\boldsymbol{X}$
- When the THC factorization is also applied to the amplitude tensor, CCSD scaling can be theoretically further reduced to $O(n^4)$
- The CP decomposition for THC can be obtained by decomposing D or by using a spatial grid
- While more effective, the latter approach does not extend to adaptations of THC to periodic systems

Library for Massively-Parallel Tensor Computations

Cyclops Tensor Framework[1] sparse/dense generalized tensor algebra

- Cyclops is a C++ library that distributes each tensor over MPI
- Used in chemistry (PySCF, QChem)[2], quantum circuit simulation (IBM/LLNL)[3], and graph analysis (betweenness centrality)[4]
- Summations and contractions specified via Einstein notation

```
E["aixbjy"] += X["aixbjy"] - U["abu"]*V["iju"]*W["xyu"]
```

- Best distributed contraction algorithm selected at runtime via models
- Support for Python (`numpy.ndarray` backend), OpenMP, and GPU
- Simple interface to core ScaLAPACK matrix factorization routines

[1] https://github.com/cyclops-community/ctf

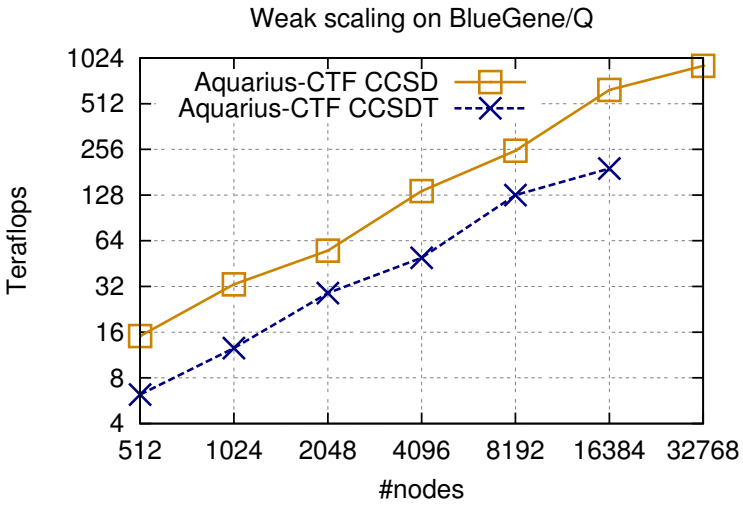
[2] E.S., D. Matthews, J. Hammond, J.F. Stanton, J. Demmel, JPDC 2014

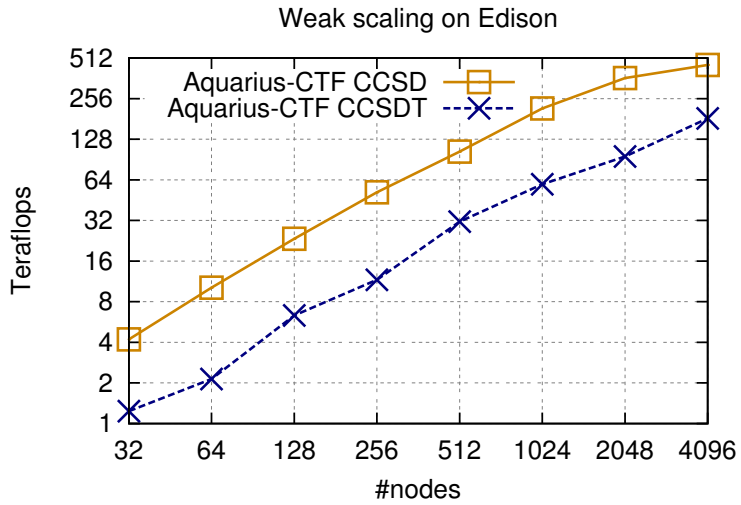
[3] E. Pednault, J.A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. S., E. Draeger, E. Holland, and R. Wisnieff, 2017

[4] E.S., M. Besta, F. Vella, T. Hoefler, SC 2017

Electronic structure calculations with Cyclops

CCSD up to 55 (50) water molecules with cc-pVDZ
CCSDT up to 10 water molecules with cc-pVDZ

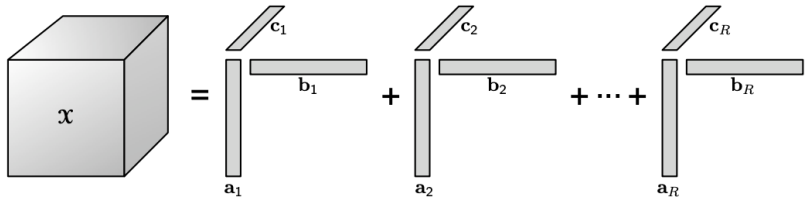




compares well to NWChem (up to 10x speed-ups for CCSDT)

Tensor Decompositions

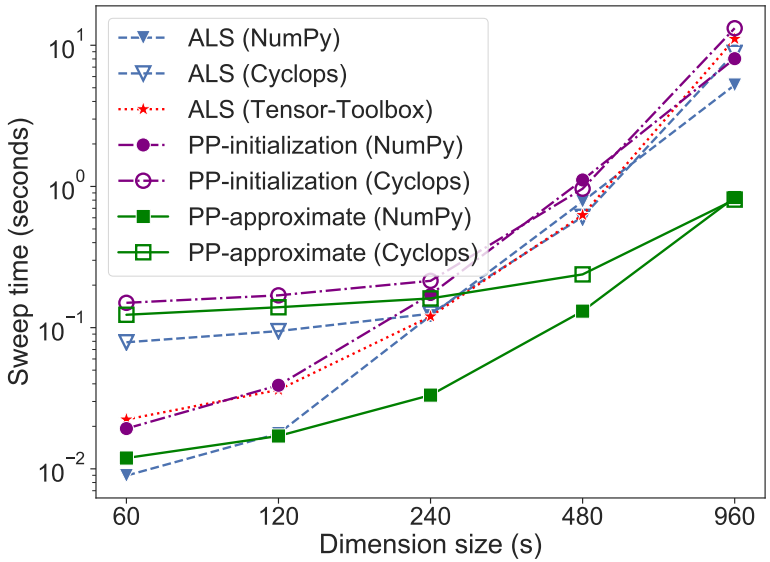
- Tensor of order N has N modes and dimensions $s \times \cdots \times s$
- Canonical polyadic (CP) tensor decomposition[1]

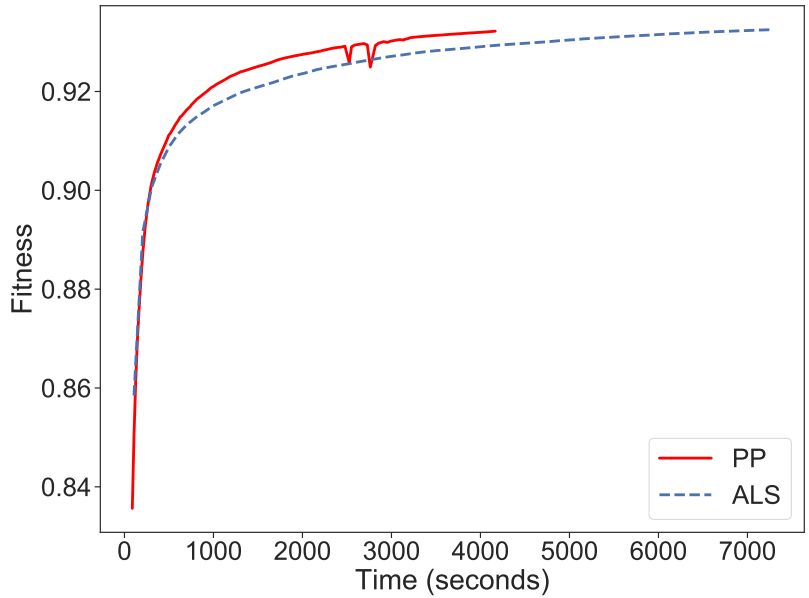


- Alternating least squares (ALS) is most widely used method
  - Monotonic linear convergence
- Gauss-Newton method is an emerging alternative
  - Non-monotonic, but can achieve superlinear convergence rate

[1] T.G. Kolda and B.W. Bader, SIAM Review 2009

Accelerating Alternating Least Squares





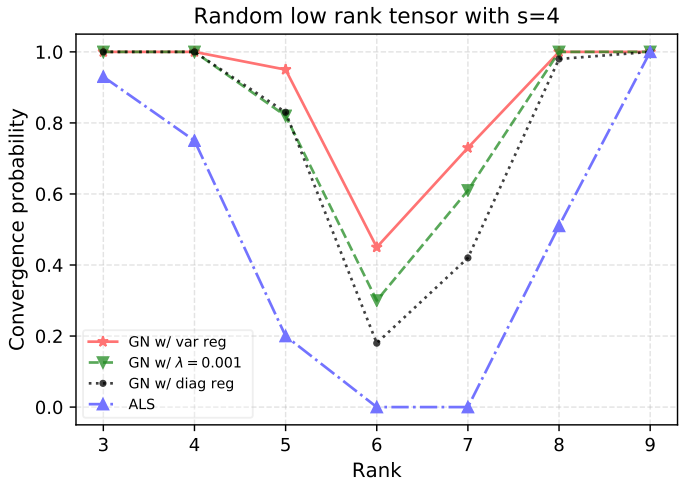
New algorithm: pairwise perturbation (PP)[1] approximates ALS

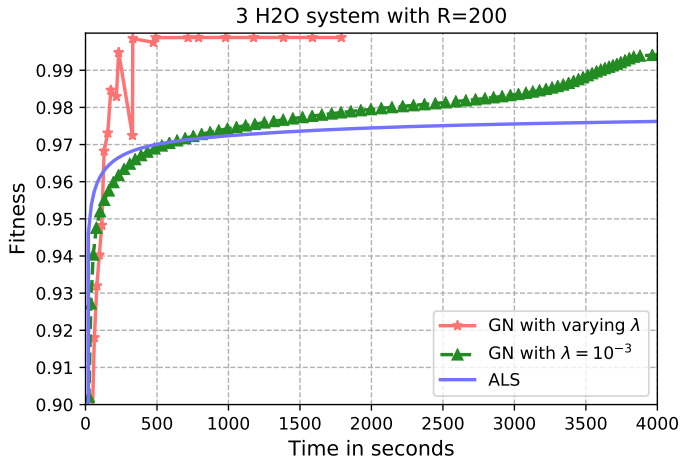
- based on perturbative expansion of ALS update
- approximation is accurate when ALS updates stagnate
- rank $R < s^{N-1}$ CP decomposition:
  - ALS sweep cost $O(s^N R) \Rightarrow O(s^2 R)$, up to 33x speed-up

Linjian Ma

[1] L. Ma, E.S. arXiv:1811.10573

Regularization and Parallelism for Gauss-Newton





New regularization scheme[1] for Gauss-Newton CP with implicit CG[2]

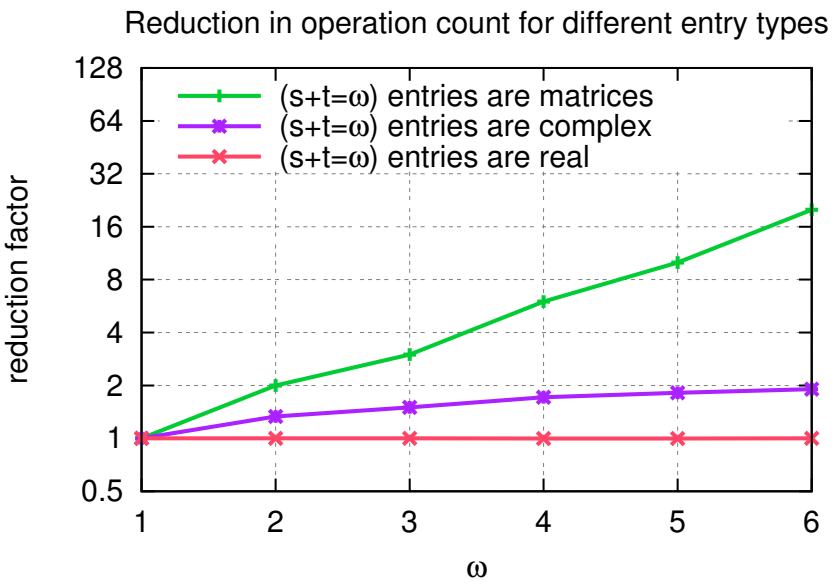
- Oscillates regularization parameter geometrically between lower and upper thresholds
- Achieves higher convergence likelihood
- More accurate than ALS in applications
- Faster than ALS sequentially and in parallel

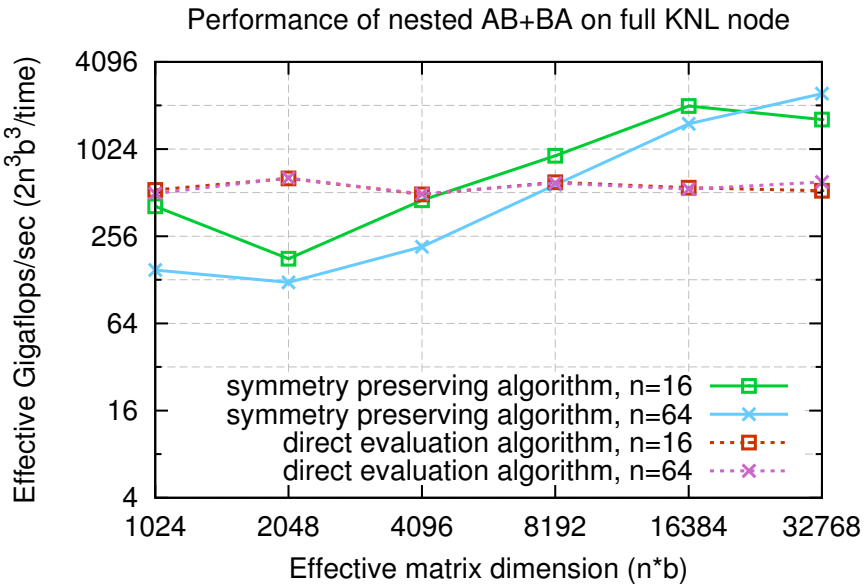
Navjot Singh

[1] Navjot Singh, Linjian Ma, Hongru Yang, and E.S. arXiv:1910.12331

[2] P. Tichavsky, A. H. Phan, and A. Cichocki., 2013

Permutational Symmetry in Tensor Contractions





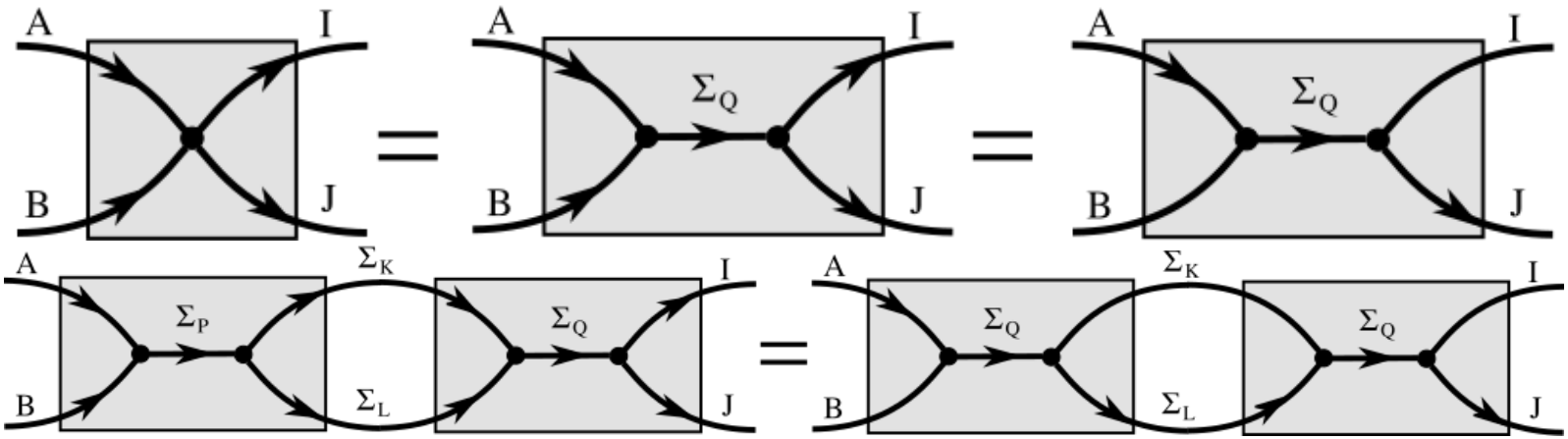
New contraction algorithms reduce cost via permutational symmetry[1]

- Symmetry is hard to use in contraction e.g. $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with $\boldsymbol{A}$ symmetric
- For contraction of order $s+v$ and $v+t$ tensors to produce an order $s+t$ tensor, previously known approaches reduce cost by $s!t!v!$
- New algorithm reduces number of *products* by $\omega!$ where $\omega = s+t+v$, leads to same reduction in *cost* for partially-symmetric contractions

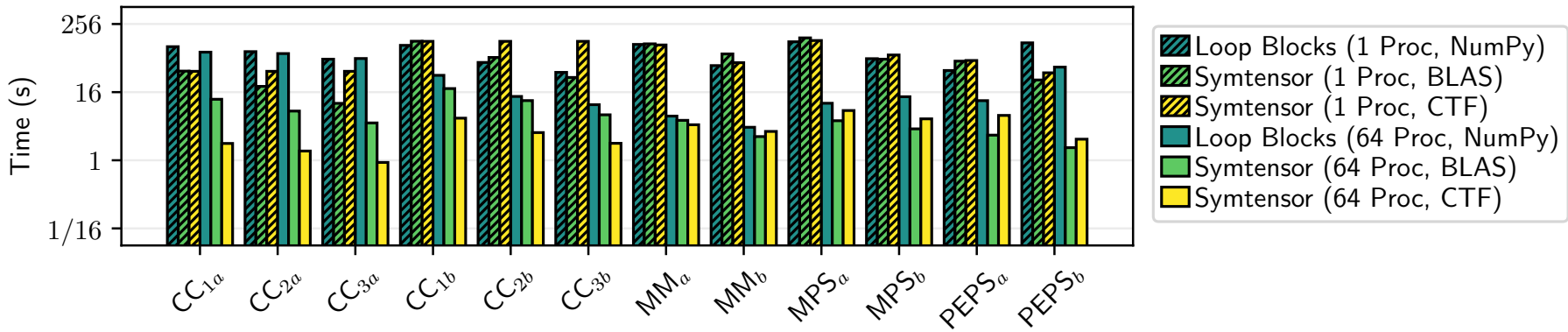
$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B} + \boldsymbol{B}\boldsymbol{A} \Rightarrow c_{ij} = \sum_k [(a_{ij} + a_{ik} + a_{jk}) \cdot (b_{ij} + b_{ik} + b_{jk})] - \ldots$$

[1] E.S, J. Demmel, CMAM 2020

Group Symmetry in Tensor Contractions



New contraction algorithm, *irreducible representation alignment* uses new reduced form to handle group symmetry (momentum consrevation, spin, quantum numbers, etc.) without looping over blocks or sparsity[1]



[1] collaboration with Yang Gao, Phillip Helms, and Garnet Chan at Caltech, to appear on arxiv, July 2020

Hamiltonians as Tensor Network Operators

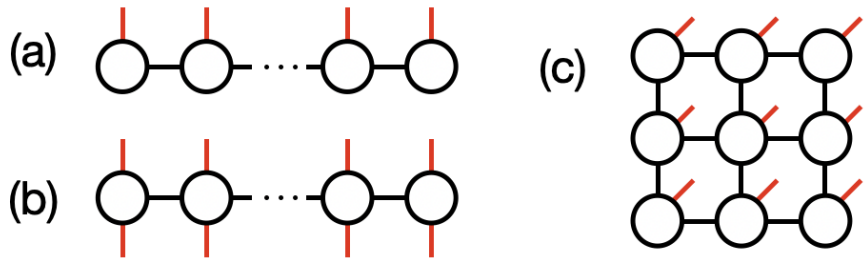
- Tensor network methods pose an alternative to Hartree-Fock-based methods for quantum chemistry
- These methods are most natural for lattice spin systems such as the Heisenberg model and the simpler transverse field Ising model

$$H = \sum_{\langle i\, j\rangle} J^z Z_i Z_j + \sum_i h_x X_i$$

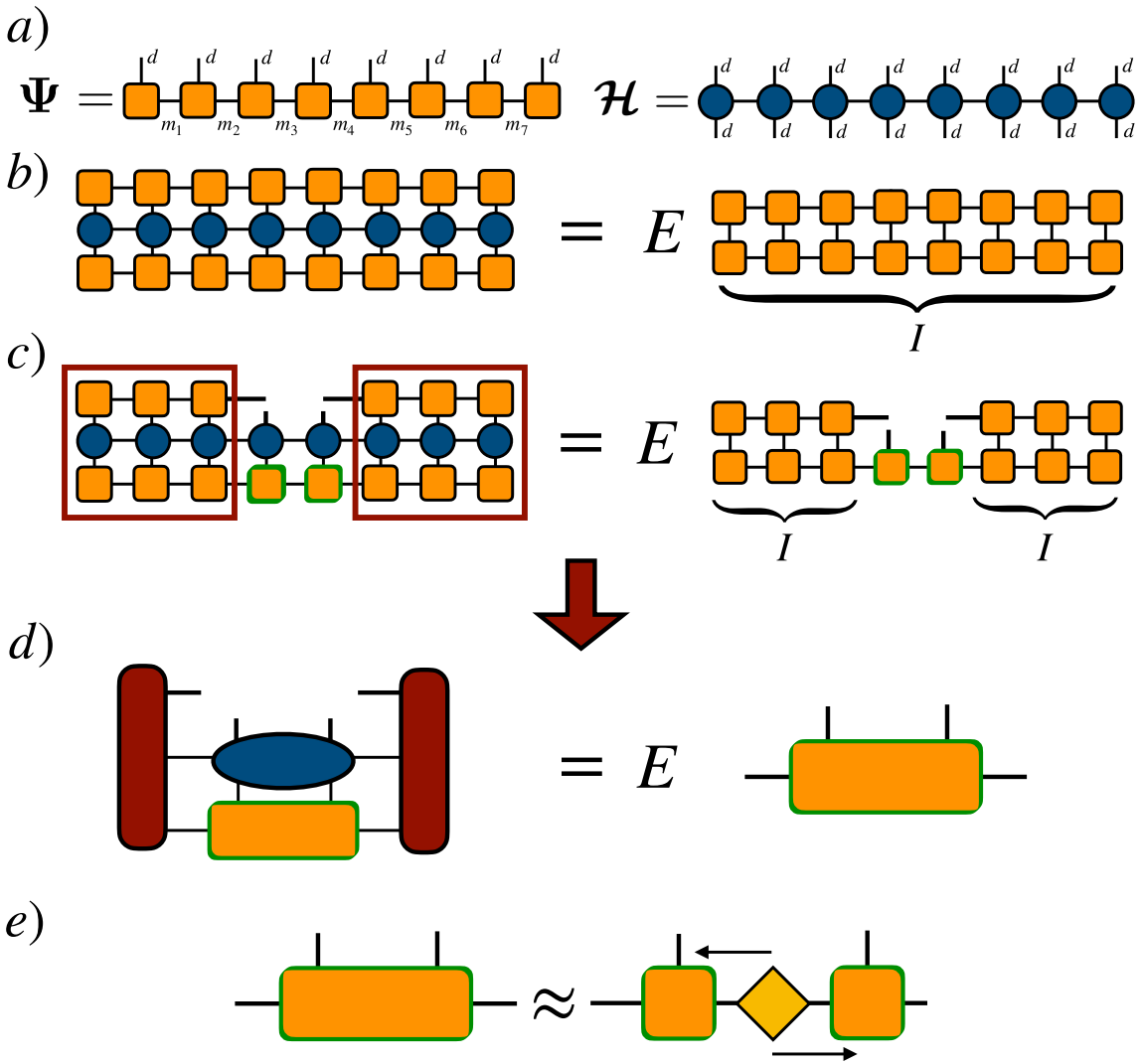
  where $\langle i\, j\rangle$ denote neighboring sites on a 2D lattice
- In the 1D case, 2-qubit operators such as $Z_i Z_{i+1}$ can be written as

$$\boldsymbol{H} = \boldsymbol{Z} \otimes \boldsymbol{Z} \otimes \boldsymbol{I} \otimes \cdots \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{Z} \otimes \boldsymbol{Z} \otimes \boldsymbol{I} \otimes \cdots \otimes \boldsymbol{I} + \cdots$$

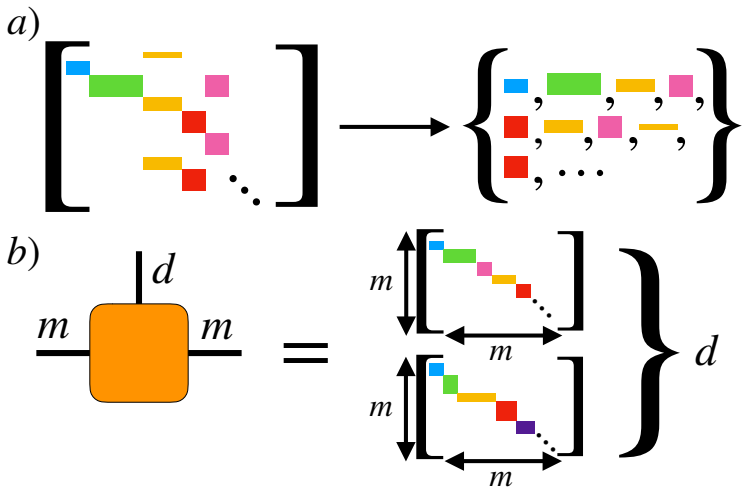
- In the 1D case, H can be represented as a matrix-product operator (MPO) with constant *bond dimension* (rank)

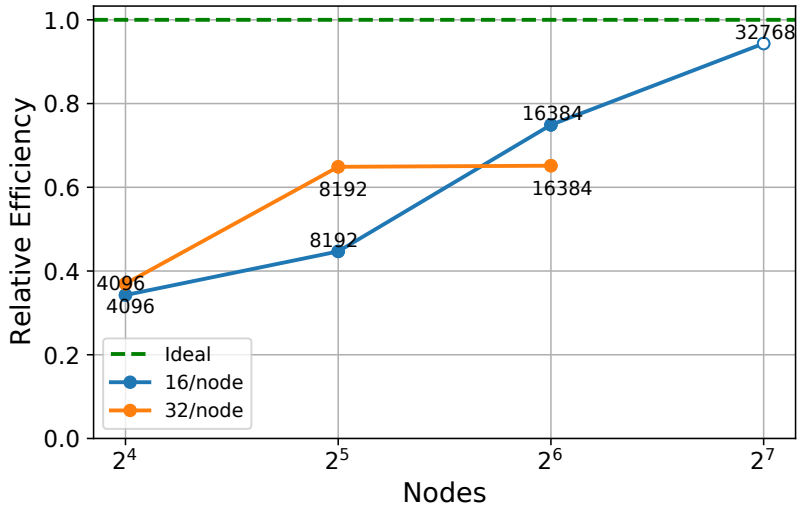


Density Matrix Renormalization Group (DMRG)



Parallel DMRG with Group Symmetry



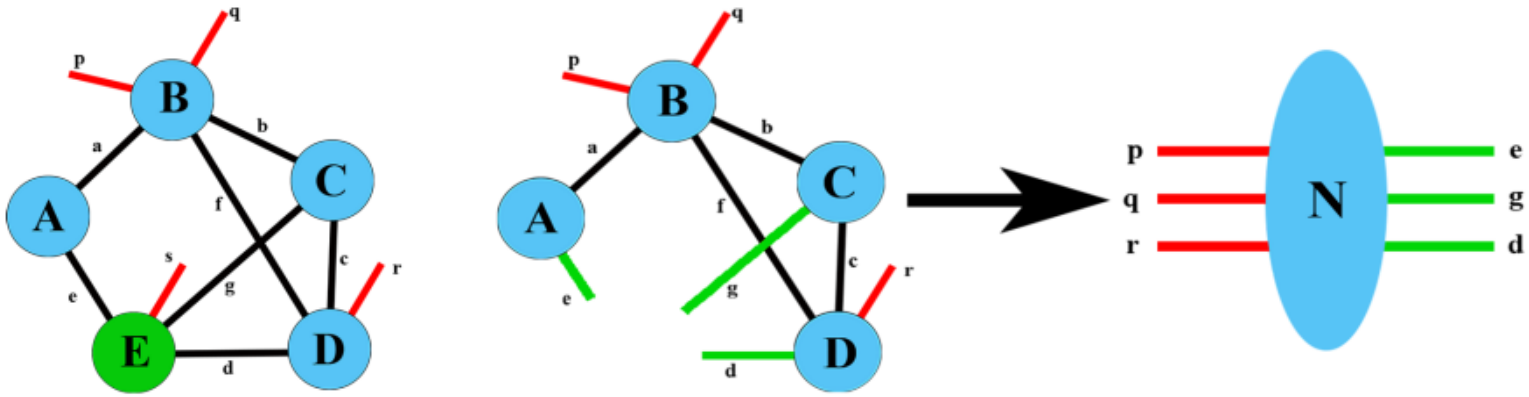


We have recently developed a parallel DMRG code using Cyclops[1]

- compare two approaches to group symmetry
  - iterate over block-wise contractions
  - use CTF's sparse tensor representation
- match ITensor efficiency at scale for spin-system, but significantly lower efficiency for fermionic system with large number of blocks

[1] collaboration with Ryan Levy and Bryan Clark (UIUC), paper to appear in proceedings of SC 2020, arXiv preprint to be released July 2020

Conditioning and Stability of Tensor Networks



(a) Tensor Network (b) Environment Matrix

- DMRG and ALS optimize one tensor at a time relative to an environment matrix (the contraction of the rest of the tensor network)
- *Canonical forms* ensure that the environment matrix is orthogonal, minimizing amplification of sitewise approximation error
- Our provides a nound on error amplification based on environment matrix condition number[1], hints at alternative approaches to ensure stability when canonical forms are hard to compute (e.g. for PEPS)

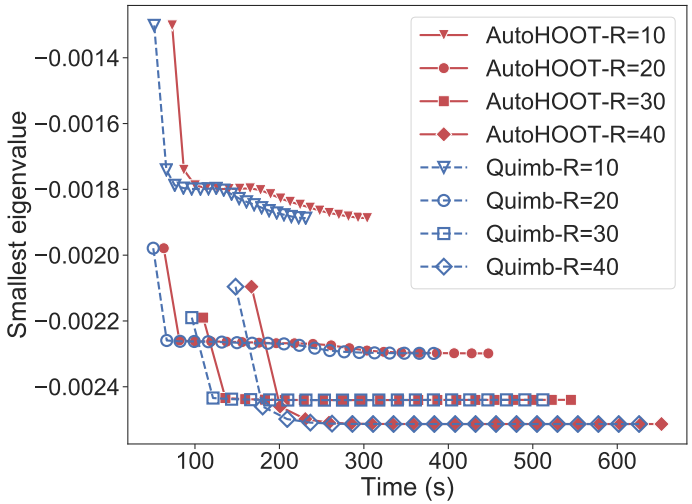
[1] Yifan Zhang and E.S. arXiv:2001.01191, 2020

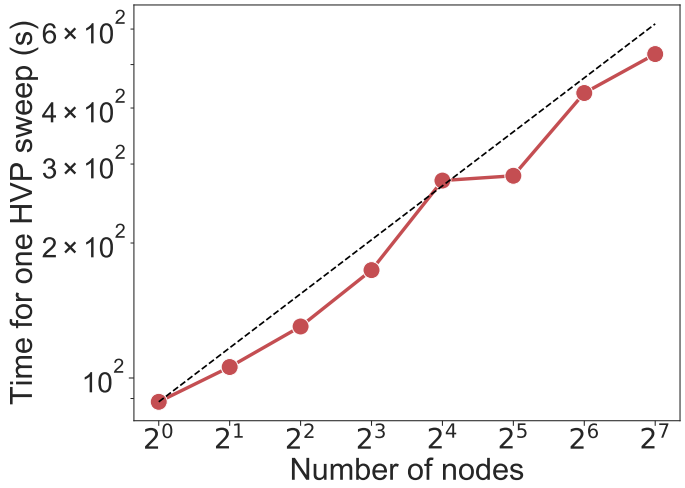
Automatic Generation of Tensor Network Methods

- Note similarity between DMRG and alternating least squares for CP decomposition
- Both apply Newton's method on a sequence of subsets of variables
- Automatic differentiation (AD) in principle enables automatic generation of these methods
- However, existing AD tools such as Jax (used by TensorFlow) are designed for deep learning and are ineffective for more complex tensor computations
  - focus purely on first order optimization via Jacobian-vector products
  - unable to propagate tensor algebra identities such as $(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} \otimes \boldsymbol{B}^{-1}$ to generate efficient code

AutoHOOT: Automatic High-Order Optimization for Tensors

- AutoHOOT[1] provides a tensor-algebra centric AD engine
- Designed for `einsum` expressions and alternating minimization common in tensor decomposition and tensor network methods
- Python-level AD is coupled with optimization of contraction order and caching of intermediates
- Generates code for CPU/GPU/supercomputers using high-level back-end interface to tensor contractions





[1] Linjian Ma, Jiayu Ye, and E.S. arXiv:2005.04540, 2020

Tensor Network State Evolution

- We can evolve a tensor network state by Trotterization of a Hamiltonian with m local terms

$$e^{-iH\tau} = \prod_{j=1}^{m} e^{-iH_j\tau} + O(\tau^2)$$

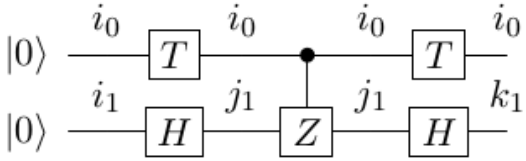
- Dynamics may be simulated by time-evolution $|\psi^{t+\tau}\rangle = e^{-iH\tau}|\psi^t\rangle$
- Ground state calculation can be done via imaginary time evolution, $|\psi^{i(t+\tau)}\rangle = e^{-H\tau}|\psi^{it}\rangle$, maximizing as follows

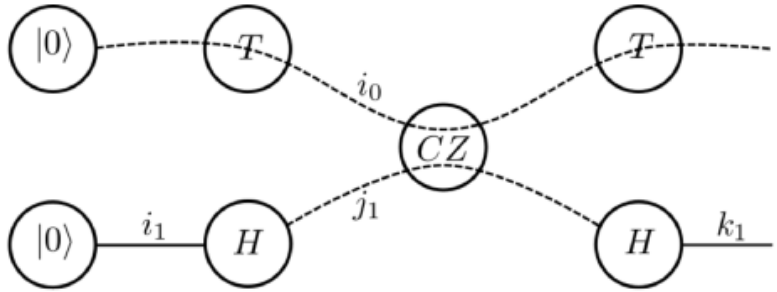
$$e^{-E\tau} = \max_{\|\psi\|_2} \langle\psi|e^{-H\tau}|\psi\rangle$$

  which is equivalent to minimizing E and leads to the same maximizer/minimzer ψ
- If H_j is a local (e.g., one/two-site) operator, so is $e^{-iH_j\tau}$

Quantum Circuit Simulation with Tensor Networks

- Evolution of tensor network states can also simulate quantum circuits
- In fact, a quantum circuit is a direct description of a tensor network[1]





- Why use HPC to (approximately) simulate quantum circuits?
  - enable development/testing/tuning of larger quantum circuits
  - understand approximability of different quantum algorithms
  - quantify sensitivty of algorithms to noise/error
  - potentially enable new hybrid quantum-classical algorithms
- Cyclops utilized to simulate 49-qubit circuits by IBM+LLNL team via direct contraction[2] and by another team from via exact PEPS evolution/contraction[3]

[1] Markov and Shi SIAM JC 2007
[2] Pednault et al. arXiv:1710.05867
[3] Guo et al. Phys Rev Letters, 2019

Tensor Formalism for Quantum Circuits

- The *state* $|\psi\rangle$ of a quantum computer with n qubits can be described by a unit vector in $\mathbb{C}^{2^n}$.
- By choosing 2^n orthonormal basis vectors/states to be denoted as $|\boldsymbol{i}\rangle$ with $\boldsymbol{i} = i_1 \cdots i_n \in \{0,1\}^n$, $|\psi\rangle$ can be written as

$$|\psi\rangle = \sum_{\boldsymbol{i} \in \{0,1\}^n} t_{\boldsymbol{i}}^{(\psi)} |\boldsymbol{i}\rangle$$

- A *single qubit gate* $G^{(k)}$ acting on the kth qubit gives

$$|\phi\rangle = G^{(k)} |\psi\rangle \;\Rightarrow\; t_{\boldsymbol{i}}^{(\phi)} = \sum_{j_k=0}^{1} g_{i_k j_k}^{(k)} t_{i_1 \cdots i_{k-1} j_k i_{k+1} \cdots i_n}^{(\psi)}$$

- A *2-qubit gate* $G^{(k,l)}$ acting on qubits k, l with $k < l$ gives

$$|\phi\rangle = G^{(k,l)} |\psi\rangle \;\Rightarrow\; t_{\boldsymbol{i}}^{(\phi)} = \sum_{j_k=0}^{1} \sum_{j_l=0}^{1} g_{i_k i_l j_k j_l}^{(k,l)} t_{i_1 \cdots j_k \cdots j_l \cdots i_n}^{(\psi)}$$

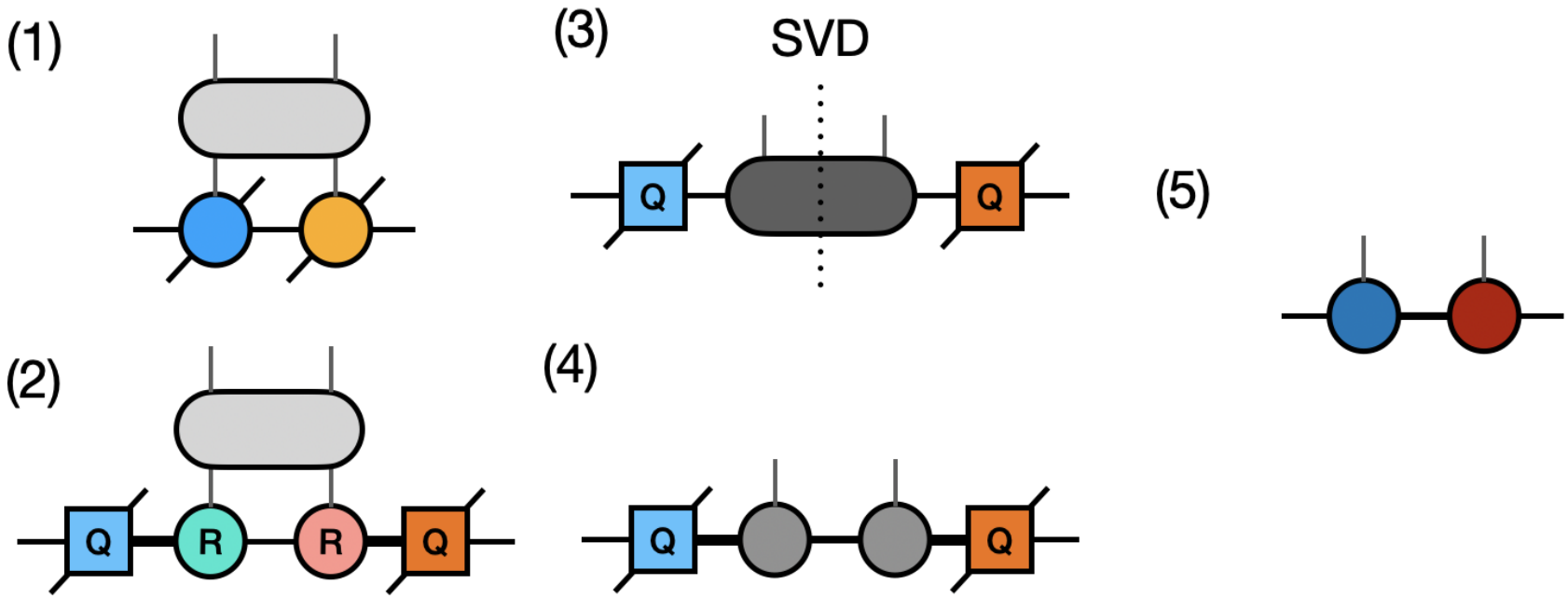
Quantum Circuit Simulation using PEPS[1]

- Near-term quantum architectures mostly connect qubits in a 2D fashion
- Non-local gates can be applied via the use of swap gates (with corresponding overhead)
- 2D tensor networks (projected entangles pair states (PEPS)) provide a natural way to simulate 2D quantum circuits
- Same software/algorithms infrastructure is also effective for (imaginary) time evolution with many Hamiltonians of interest
- Gate application and contraction of PEPS can both have exponential cost in the size of the circuit, so desire effective approximation

[1] Yuchen Pang, Tianyi Hao, Annika Dugad, Yiqing Zhou, and E.S., to appear in proceedings of SC 2020, arXiv:2006.15234.

Approximate Application of Two-Site Operators

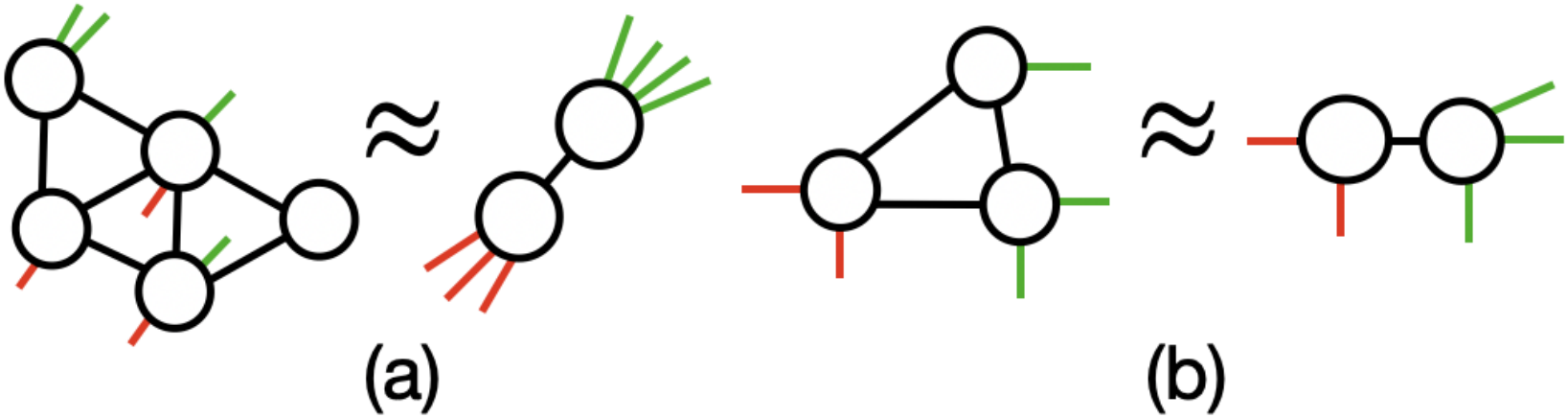
- Consider application of a two-site operator on neighboring PEPS sites
- *Simple update (QR-SVD)* algorithm:



- We provide an efficient distributed implementation of QR-SVD
- This operation is an instance of what we'll refer to as `einsumsvd` and QR-SVD is one algorithm/implementation

Implicit Randomized `einsumsvd`

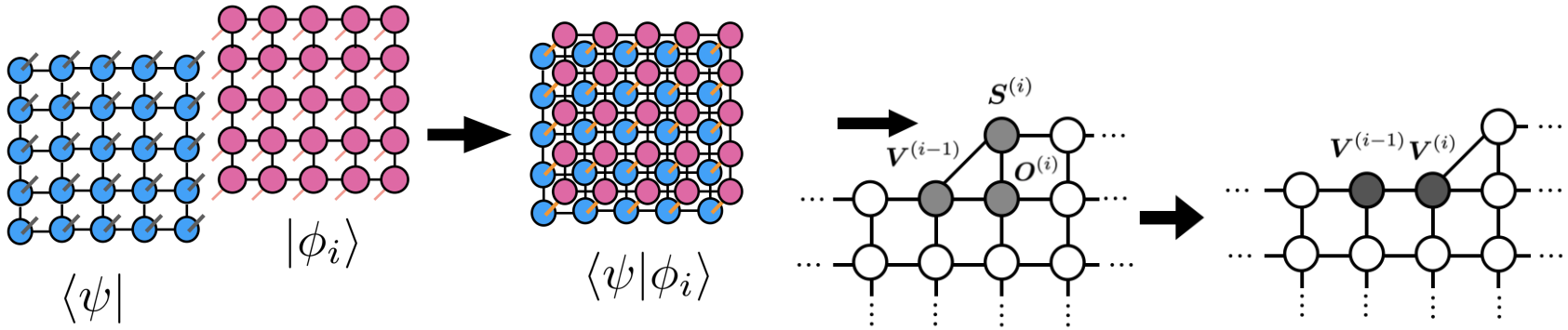
- The `einsumsvd` primitive will also enable effective algorithms for PEPS contraction



- An efficient general implementation is to leverage randomized SVD / orthogonal iteration, which iteratively computes a low-rank SVD by a matrix–matrix product that can be done implicitly via tensor contractions

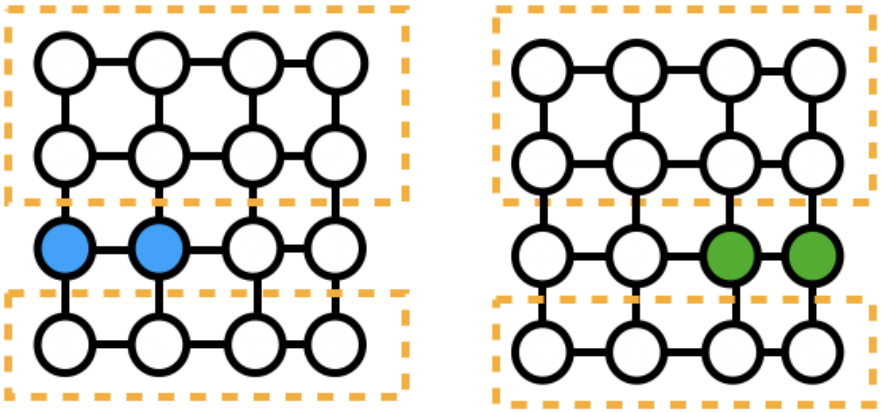
PEPS Contraction

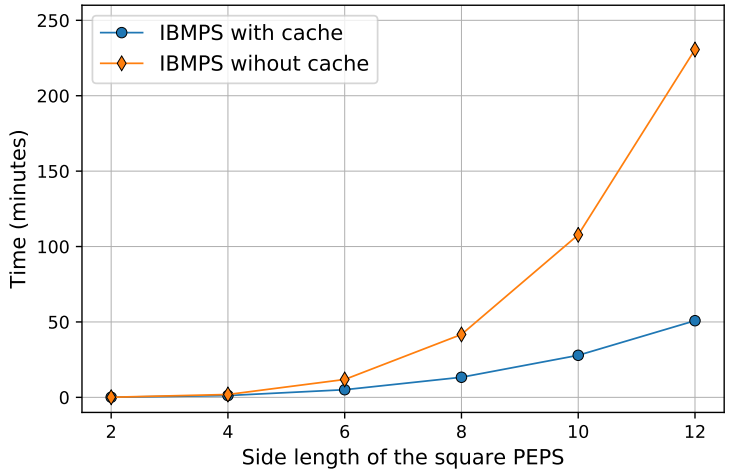
- Exact contraction of PEPS is #P-complete, so known methods have exponential cost in the number of sites
- PEPS contraction is needed to compute expectation values such as $\langle\psi| H |\psi\rangle$
- *Boundary contraction* is common for finite PEPS and can be simplified with `einsumsvd`



Computing Expectation Values with PEPS

- To compute $\langle\psi| H |\psi\rangle$, we could compute each $\langle\psi| H_i |\psi\rangle$ and sum
- To improve performance, leverage caching of intermediates accross different expectation values of local operators



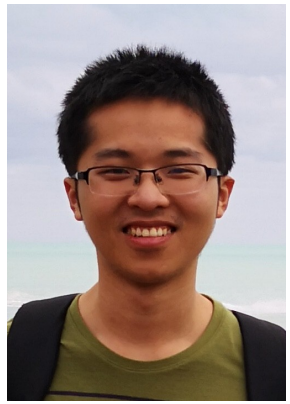


- An alternative efficient implementation can be obtained by computing the expectation value of the time-evolution operator $e^{-iH\tau}$
- Caching approach also enables computation of unsummed expectation values, which are useful for gradients (needed in e.g., Adapt-VQE)

Koala

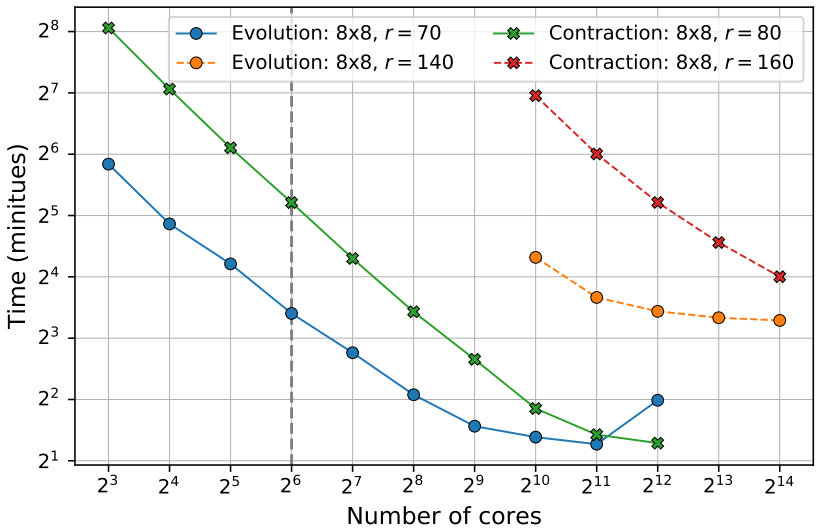
- We introduce a new library, Koala[1], for high-performance simulation of quantum circuits and time evolution with PEPS

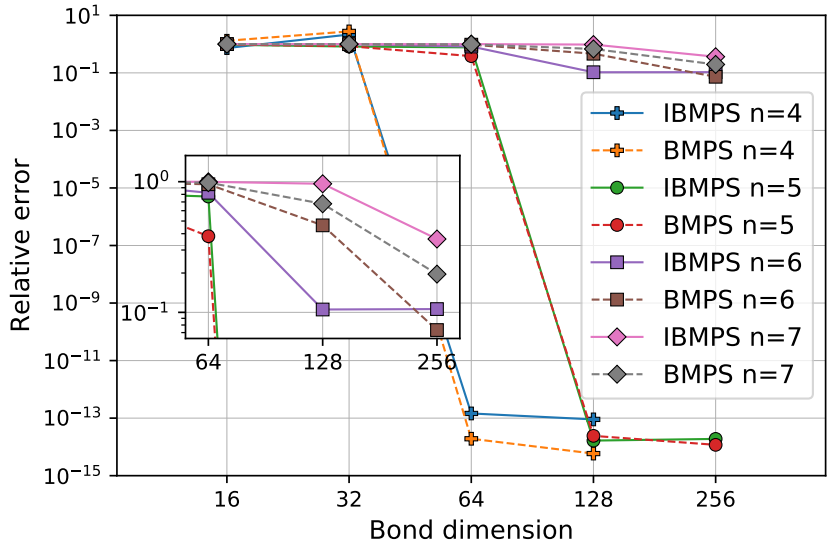
```python
from koala import peps, Observable
from tensorbackends.interface import ImplicitRandomizedSVD

# Create a 2-by-3 PEPS in distributed memory using CTF
qstate = peps.computational_zeros(nrow=2, ncol=3, backend='ctf')

# Construct operators and apply them to the quantum state
Y = qstate.backend.astensor([0,-1j,1j,0]).reshape(2,2)
CX = qstate.backend.astensor([1,0,0,0,0,1,0,0,0,0,0,1,0,0,1,0,])
CX = CX.reshape(2,2,2,2)

qstate.apply_operator(Y, [1])
qstate.apply_operator(CX, [1,4], update_option=peps.QRUpdate(rank=2))

# Construct an observable and calculate the expectation with IBMPS
observable = Observable.ZZ(3, 4) + 0.2 * Observable.X(1)
result = qstate.expectation(
    observable, use_cache=True,
    contract_option=peps.BMPS(ImplicitRandomizedSVD(rank=4)),
)
```



[1] https://github.com/cyclops-community/koala

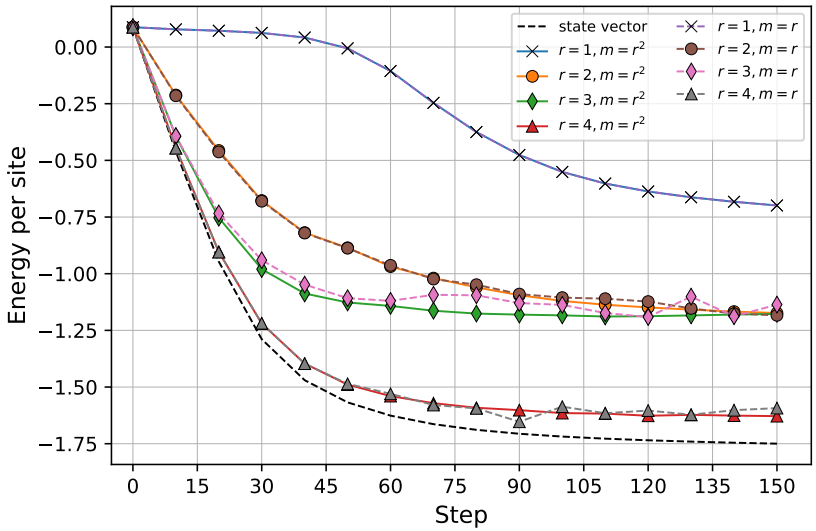
PEPS Benchmark Performance

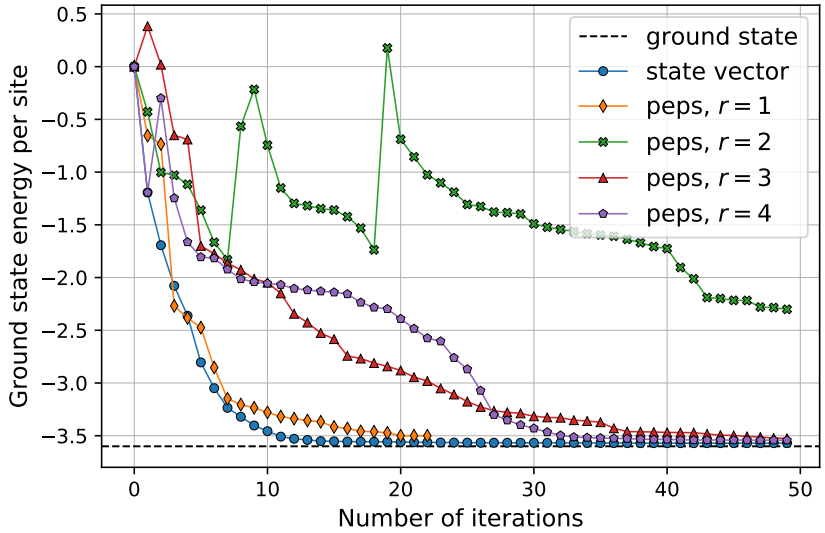




- Koala achieves good parallel scalability for approxmate gate application (evolution) and contraction
- Approximation can be effective even for adversarially-designed circuits such as Google's random quantum circuit model (figure on right)

PEPS Accuracy for Quantum Simulation





- ITE code achieves improvable accuracy with increased PEPS bond dimension, but approximation in PEPS contraction is not variational
- Variational quantum eigensolver (VQE), which represents a wavefunction using a parameterized circuit $U(\theta)$ and minimizes

$$\langle U(\theta)| H |U(\theta)\rangle ,$$

also achieves improvable accuracy with higher PEPS bond dimension

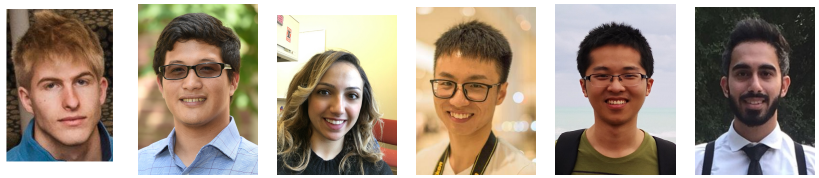
Conclusion

- Our research group is developing an ecosystem of algorithms and software for simulation of quantum systems
- This work is relevant to both classical methods for quantum chemistry and physics, as well as quantum computation

Acknowledgements

- Laboratory for Parallel Numerical Algorithms (LPNA) at University of Illinois, `lpna.cs.illinois.edu` and collaborators
- Funding from NSF awards: #1839204 (RAISE-TAQS), #1931258 (CSSI), #1942995 (CAREER)
- Stampede2 resources at TACC via XSEDE

L·P·N·A @CS@Illinois



DOE CSGF

NSF

`http://lpna.cs.illinois.edu`