

Scalable Algorithms for Tensor Computations

Edgar Solomonik

 @CS@Illinois

Department of Computer Science
University of Illinois at Urbana-Champaign

Fast Code Seminar, MIT

Laboratory for Parallel Numerical Algorithms

Recent/ongoing research topics

- parallel matrix computations
 - QR factorization
 - triangular solve
 - eigenvalue problems
- 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
 - fast matrix multiplication



L · P · N · A @ CS @ Illinois



<http://lpna.cs.illinois.edu>

Outline

- 1 Introduction
- 2 Cyclops
- 3 Tensor Decompositions
- 4 Tensor Networks
- 5 Fast Bilinear Algorithms
- 6 Conclusion

Library for Massively-Parallel Tensor Computations

Cyclops Tensor Framework¹ 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

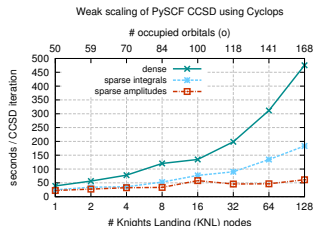
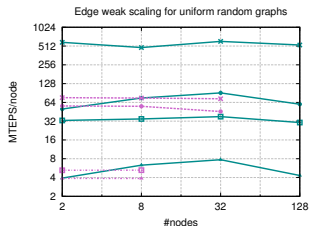
¹<https://github.com/cyclops-community/ctf>

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

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

⁴E.S., M. Besta, F. Vella, T. Hoefer, SC 2017

Sparsity in Tensor Contractions



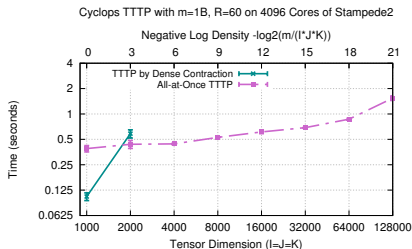
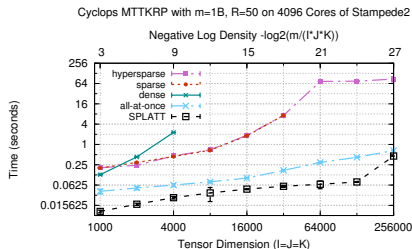
- Cyclops supports sparse representation of tensors¹
- Choice of representation specified in tensor constructor
- CSR or DCSR² (2-index CSF³) representation used locally for contractions

¹E.S., T. Hoeffler 2015

²A. Bulúć, J.R. Gilbert, 2008

³S. Smith, G. Karypis 2015

All-at-Once Multi-Tensor Contraction



With sparsity, all-at-once contraction¹ of multiple tensors can be faster².

- Sparse CP decomposition methods dominated in cost by MTTKRP

$$u_{ir} = \sum_{j,k} t_{ijk} v_{jr} w_{kr}$$

- All-at-once sparse MTTKRP needs less communication than pairwise

- Tensor times tensor product (TTTP) enables sparse residual and CP tensor completion

$$r_{ijk} = \sum_r t_{ijk} u_{ir} v_{jr} w_{kr}$$

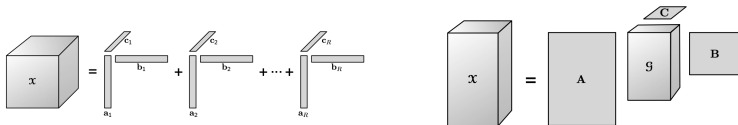
- Cost and memory footprint reduced asymptotically

¹ S. Smith, J. Park, G. Karypis, 2018

² Zecheng Zhang, Xiaoxiao Wu, Naijing Zhang, Siyuan Zhang, and E.S. arXiv:1910.02371

CP Tensor Decomposition Algorithms

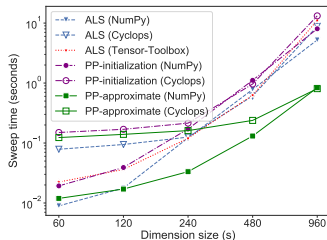
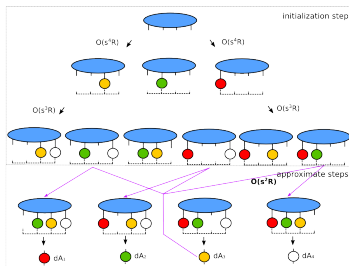
- Tensor of **order** N has N **modes** and **dimensions** $s \times \cdots \times s$
- **CP** and **Tucker** tensor decompositions¹



- Alternating least squares (**ALS**) is most widely used method
 - Optimize one factor matrix at a time, yielding quadratic optimization subproblems
 - Achieves monotonic linear convergence
- **Gauss-Newton** method is an emerging alternative
 - Optimizes all factor matrices at once by quadratic approximation of nonlinear objective function
 - Non-monotonic, but can achieve quadratic convergence

¹Kolda and Bader, SIAM Review 2009

Pairwise Perturbation Algorithm



New algorithm: **pairwise perturbation (PP)**¹ approximates ALS

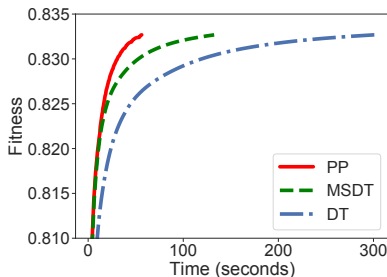
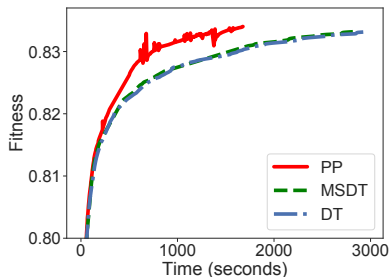
- based on perturbative expansion of ALS update to approximate MTTKRP
- approximation is accurate when ALS updates stagnate
- rank $R < s^{N-1}$ CP decomposition:
 - ALS sweep cost $O(s^N R) \Rightarrow O(s^2 R)$, up to 33x speed-up



Linjian Ma

¹Linjian Ma, E.S. arXiv:1811.10573

Parallel Pairwise Perturbation Algorithm



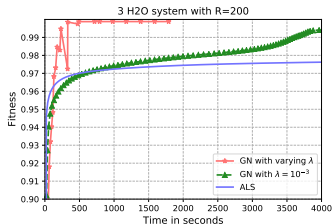
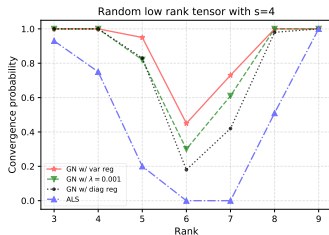
Effective parallelization by decomposing MTTKRP into local MTTKRP_s¹

$$U = \text{MTTKRP}(\mathcal{T}, V, W) \Rightarrow U_i = \sum_{j,k} \text{MTTKRP}(\mathcal{T}_{ijk}, V_j, W_k)$$

- processor (i, j, k) owns $\mathcal{T}_{ijk}$, V_j , and W_k
- pairwise perturbation can be used to approximate local MTTKRP_s
- multi-sweep dimension-tree (MSDT) amortizes terms across sweeps

¹Linjian Ma, E.S. IPDPS 2021

Regularization and Parallelism for Gauss-Newton



New regularization scheme¹ for Gauss-Newton CP with implicit CG²

- 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

Randomized Methods for Sparse Tensor Decomposition

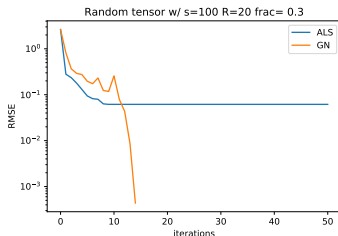
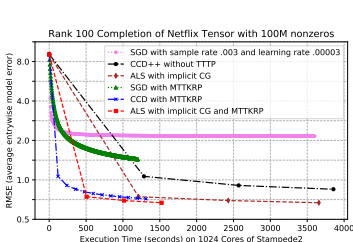
- When seeking a low-rank $R = O(1)$ decomposition for a sparse tensor, sketching schemes have been shown to be efficient
- In this regime, Tucker can be used to construct a CP decomposition
- Leverage score sampling on the rank-constrained least squares problem $\min_{\mathbf{X}, \text{rank}(\mathbf{X}) \leq R} \|\mathbf{A}\mathbf{X} - \mathbf{B}\|_F$ leads to a state-of-the-art cost-accuracy trade-off¹ in approximations to Tucker-ALS
- Ideas similar to work by Liu and Moitra (2020) on tensor completion

Algorithm for Tucker	LS solve cost	Sample size (m)
ALS	$O(\text{nnz}(\mathcal{T})R^{N-1})$	/
ALS + TensorSketch ²	$\tilde{O}(mR^N + msR)$	$O(R^{2(N-1)} \cdot 3^{N-1}/(\epsilon^2\delta))$
ALS + TTMTS ²	$\tilde{O}(msR^{N-1})$	$O(R^{2(N-1)} \cdot 3^{N-1}/(\epsilon^2\delta))$
ALS + TensorSketch ¹	$\tilde{O}(mR^{2N-2} + sR^{N-1})$	$O((R^{(N-1)} + 1/\epsilon^2) \cdot (3R)^{(N-1)}/\delta)$
ALS + leverage scores ¹	$\tilde{O}(mR^{2N-2} + sR^{N-1})$	$O(R^{(N-1)}/(\epsilon^2\delta))$

¹ Linjian Ma and E.S., in preparation

² O. Malik and S. Becker, 2018 (assuming unconstrained LSQ)

Tensor Completion

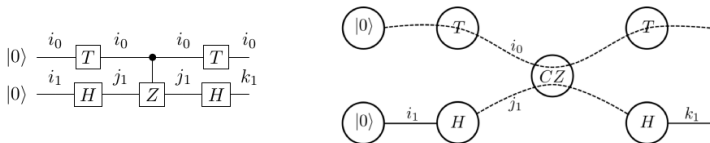


- Via the CTF Python interface, we have implemented SGD, CCD, ALS (with iterative and direct solves), and Gauss-Newton¹
- Can also handle a variety of loss functions (generalized decomposition)
- All-at-once primitives for MTTKRP, TTTP, and explicit solves in completion ALS drastically improve performance

¹Navjot Singh, Zecheng Zhang, Xiaoxiao Wu, Naijing Zhang, Siyuan Zhang, and Edgar Solomonik arXiv:1910.02371

Quantum Circuit Simulation with Tensor Networks

- A quantum circuit is a direct description of a tensor network¹



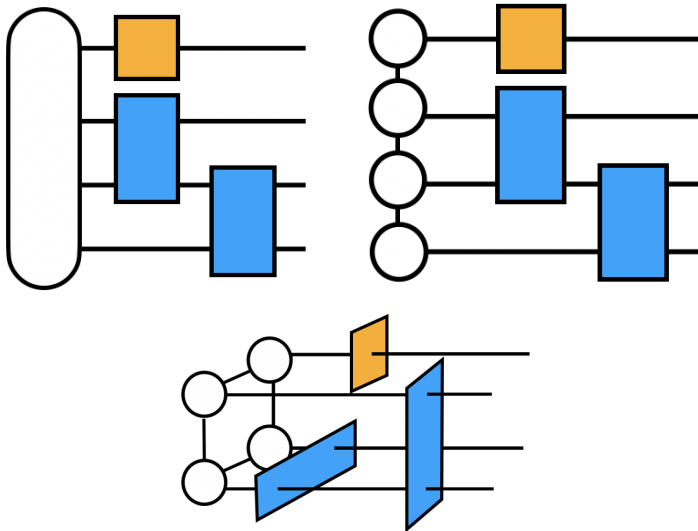
- Why use HPC to (approximately) simulate quantum circuits?
 - enable development/testing/tuning of larger quantum circuits
 - understand approximability of different quantum algorithms
 - quantify sensitivity of algorithms to noise/error
 - potentially enable new hybrid quantum-classical algorithms
- Cyclops utilized to simulate 49-qubit circuits by IBM+LLNL team via direct contraction² and by another team from via exact PEPS evolution/contraction³

¹Markov and Shi SIAM JC 2007

²Pednault et al. arXiv:1710.05867

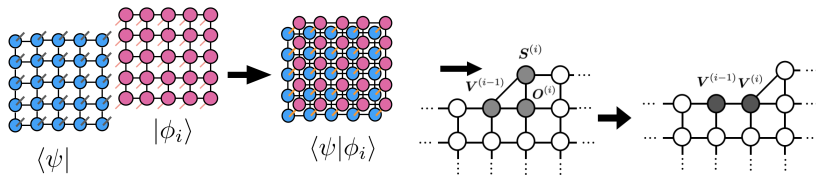
³Guo et al. Phys Rev Letters, 2019

Tensor Network State Simulation



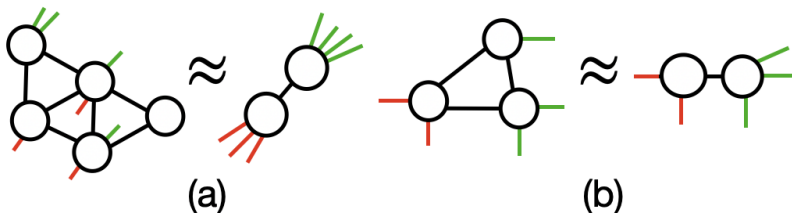
PEPS Contraction

- 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
- Boundary contraction* is common for finite PEPS and can be simplified with einsumsvd



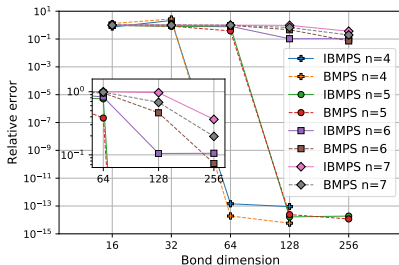
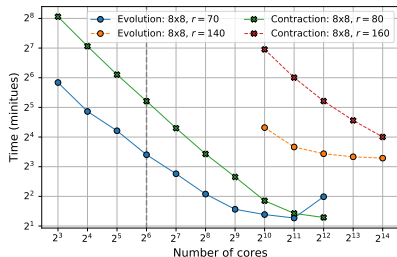
Implicit Randomized einsumsvd

- The einsumsvd primitive provides an effective abstraction for tensor network simulation methods



- 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 Benchmark Performance

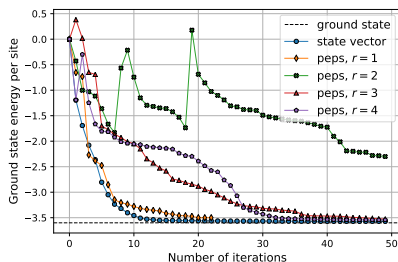
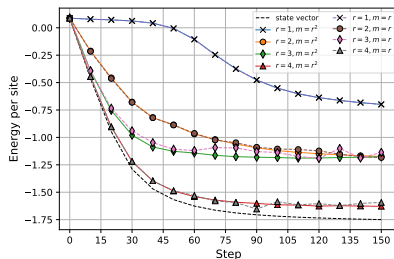


- We introduce a new library, Koala¹, for high-performance simulation of quantum circuits and time evolution with PEPS²
- Koala achieves good parallel scalability for approximate gate application (evolution) and contraction
- Approximation can be effective even for adversarially-designed circuits such as Google's random quantum circuit model (figure on right)

¹<https://github.com/cyclops-community/koala>

²Yuchen Pang, Tianyi Hao, Annika Dugad, Yiqing Zhou, and E.S. SC 2020

PEPS Accuracy for Quantum Simulation



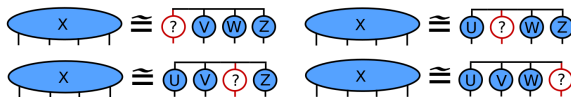
- ITE code achieves improvable accuracy with increased PEPS bond dimension, but approximation in PEPS contraction is not variational
- Variational quantum eigensolver (VQE), which represents a wavefunction using a parameterized circuit $U(\theta)$ and minimizes

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

also achieves improvable accuracy with higher PEPS bond dimension

Automatic Differentiation for Tensor Computations

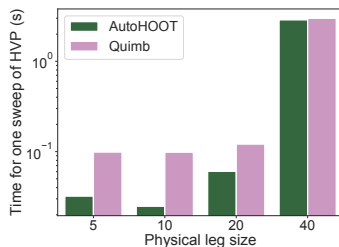
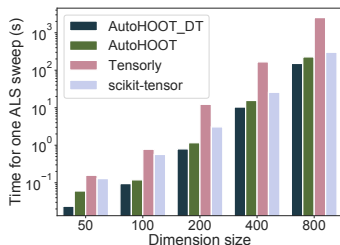
- Tensor network and tensor decomposition methods all typically based on applying Newton's method on a sequence of subsets of variables



- 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
 - these focus purely on first order optimization via Jacobian-vector products
 - unable to propagate tensor algebra identities such as $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ to generate efficient code

AutoHOOT: Automatic High-Order Optimization for Tensors

- AutoHOOT¹ 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



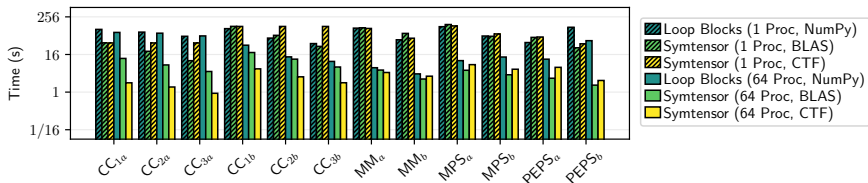
¹Linjian Ma, Jiayu Ye, and E.S. PACT 2020

Group Symmetry in Tensor Contractions

- Tensor with cyclic group symmetry can be represented as block-sparse

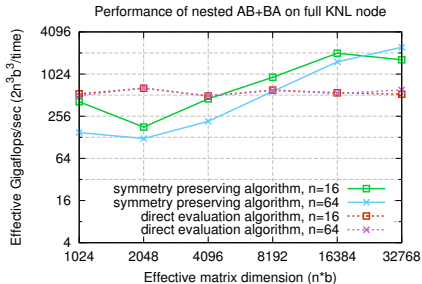
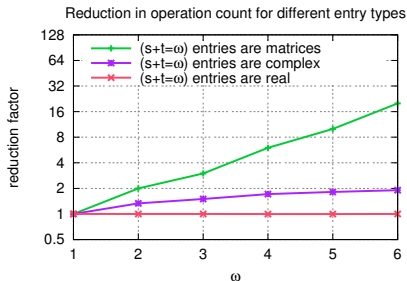
$$t_{ijk\dots} = 0 \quad \text{if} \quad \lfloor i/G_1 \rfloor + \lfloor j/G_2 \rfloor + \lfloor k/G_3 \rfloor + \dots \neq 0 \pmod{G}.$$

- Group symmetries of multiple types arise due to conservation laws when physical systems (quantum number symmetry, spin symmetry, rotational symmetry, translational symmetry)
- New contraction algorithm, *irreducible representation alignment* uses new dense reduced form tensor to handle group symmetry without looping over blocks or sparsity¹



¹Y. Gao, P. Helms, G. Chan, and E.S., arXiv:2007.08056

Permutational Symmetry in Tensor Contractions



New contraction algorithms reduce cost via permutational symmetry¹

- Symmetry is hard to use in contraction e.g. $y = Ax$ with 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

$$C = AB + BA \Rightarrow c_{ij} = \sum_k [(a_{ij} + a_{ik} + a_{jk}) \cdot (b_{ij} + b_{ik} + b_{jk})] - \dots$$

¹E.S. J. Demmel, CMAM 2020

Communication Cost of Fast Bilinear Algorithms

- Given inputs $\mathbf{a}$ and $\mathbf{b}$, a bilinear algorithm computes

$$\mathbf{c} = \mathbf{F}^{(C)}[(\mathbf{F}^{(A)T} \mathbf{a}) \circ (\mathbf{F}^{(B)T} \mathbf{b})]$$

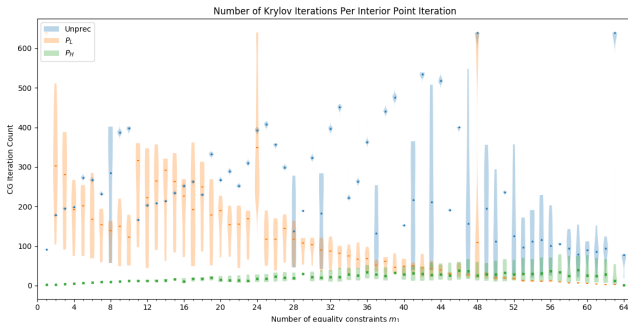
- All fast algorithms for matrix multiplication, convolution, and symmetric tensor contractions are bilinear algorithms
- Communication lower bounds can be attained for any execution of a bilinear algorithm, given a lower bound on the rank of subset of columns of $\mathbf{F}^{(A)}$, $\mathbf{F}^{(B)}$, or $\mathbf{F}^{(C)}$ ¹
- Can automatically obtain rank lower bounds for $\mathbf{A} \otimes \mathbf{B}$ from that of $\mathbf{A}$ and $\mathbf{B}$, enabling application of these lower bounds to nested algorithms such Strassen's algorithm, convolution, and symmetry preserving algorithms applied to partially symmetric contractions²

¹E.S., J. Demmel, T. Hoefler arXiv:1707.04618

²Caleb Ju, Yifan Zhang, E.S. in preparation

Conclusion and Future Directions

- Talk introduced new algorithms and software for tensor contractions, tensor decomposition, and tensor networks, considering challenges involved in handling symmetry, sparsity, and parallelism
- We are also exploring solvers for QP interior point methods via a new Schur complementation strategy and preconditioners¹



¹Samah Karim and E.S. in preparation

Acknowledgements

- Laboratory for Parallel Numerical Algorithms (LPNA) at University of Illinois
- This work has been supported by NSF awards #1839204 (RAISE-TAQS), #1931258 (CSSI), #1942995 (CAREER)
- Stampede2 resources at TACC via XSEDE



LPNA @CS@Illinois



<http://lpna.cs.illinois.edu>