Tensor Software and Algorithms for Quantum Chemistry

Edgar Solomonik

L P N A @CS@Illinois

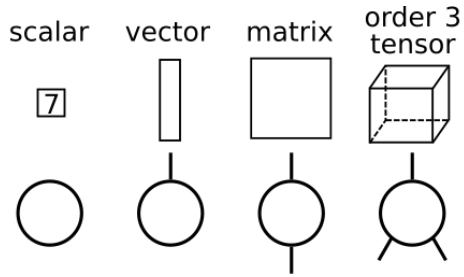
Department of Computer Science
University of Illinois at Urbana-Champaign

Quantum Chemistry Methods for Materials Science
CECAM
EPFL, Lausanne

Tensors

A tensor is a collection of elements

- its dimensions define the size of the collection
- its order is the number of different dimensions
- specifying an index along each tensor mode defines an element of the tensor



A few examples of tensors are

- Order 0 tensors are scalars, e.g., $s \in \mathbb{R}$
- Order 1 tensors are vectors, e.g., $\boldsymbol{v} \in \mathbb{R}^n$
- Order 2 tensors are matrices, e.g., $\boldsymbol{A} \in \mathbb{R}^{m \times n}$
- An order 3 tensor with dimensions $s_1 \times s_2 \times s_3$ is denoted as $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{s_1 \times s_2 \times s_3}$ with elements t_{ijk} for $i \in \{1, \ldots, s_1\}, j \in \{1, \ldots, s_2\}, k \in \{1, \ldots, s_3\}$

Tensor Contractions

A tensor contraction describes a set of products and sums of elements from two tensors

tensor contraction	formula
inner product	$w = \sum_i u_i v_i$
outer product	$w_{ij} = u_i v_{ij}$
pointwise product	$w_i = u_i v_i$
Hadamard product	$w_{ij} = u_{ij} v_{ij}$
matrix multiplication	$w_{ij} = \sum_k u_{ik} v_{kj}$
batched mat.-mul.	$w_{ijl} = \sum_k u_{ikl} v_{kjl}$
tensor times matrix	$w_{ilk} = \sum_j u_{ijk} v_{lj}$

Tensor contractions are prevalent in quantum chemistry methods

General Tensor Contractions

Given tensor $\boldsymbol{\mathcal{U}}$ of order $s+v$ and $\boldsymbol{\mathcal{V}}$ of order $v+t$, a tensor contraction summing over v modes can be written as

$$w_{i_1 \ldots i_s j_1 \ldots j_t} = \sum_{k_1 \ldots k_v} u_{i_1 \ldots i_s k_1 \ldots k_v} v_{k_1 \ldots k_v j_1 \ldots j_t}$$

- Other contractions can be mapped to this form after transposition

Unfolding tensors reduces the tensor contraction to matrix multiplication

- Combine consecutive indices in appropriate groups of size s, t, and v
- If all tensor modes are of dimension n, obtain matrix–matrix product $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$ where $\boldsymbol{C} \in \mathbb{R}^{n^s \times n^t}$, $\boldsymbol{A} \in \mathbb{R}^{n^s \times n^v}$, and $\boldsymbol{B} \in \mathbb{R}^{n^v \times n^t}$
- Assuming classical matrix multiplication, contraction requires n^{s+t+v} elementwise products and $n^{s+t+v} - n^{s+t}$ additions

Library for Massively-Parallel Tensor Contractions

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, CC4S)[2], quantum circuit simulation (by IBM/LLNL)[3], and graph analysis (betweenness centrality[4], minimum spanning tree[5])
- 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 CUDA

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

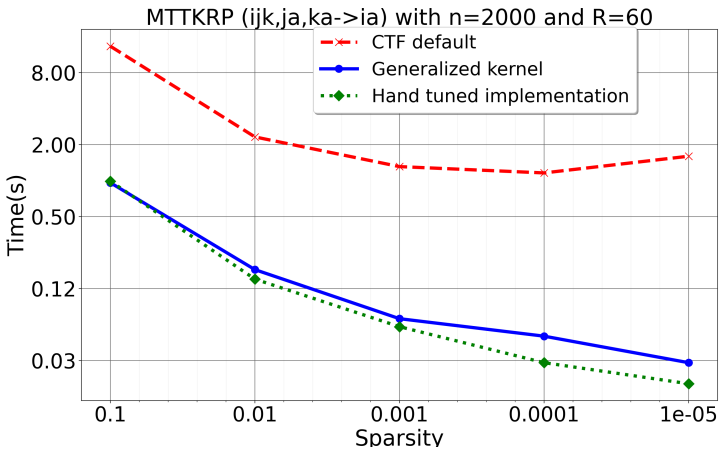
[2] E.S., D. Matthews, J. Hammond, J.F. Stanton, J. Demmel, JPDC 2014

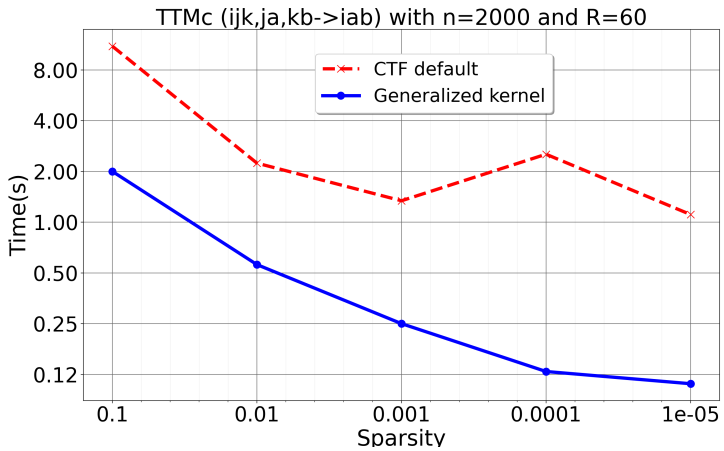
[3] E. Pednault, J.A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E.S., E. Draeger, E. Holland, and R. Wisnieff, 2017

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

[5] T. Baer, R. Kanakagiri, E.S., SIAM PP 2022

Recent and Ongoing Cyclops Developments





- All-at-once contraction for sparse tensor times many dense tensors
  - Working on integration with linear (least-squares) solves
  - Driven by tensor completion and quasi-robust density fitting[1]
- Performance models based on tensor completion
  - Given execution times $T(m, n, k)$ for $(m, n, k) \in \Omega$, predict $T(m, n, k)$ for any other (m, n, k)
  - Higher accuracy than prior art (extra trees, sparse grid regression)

[1] D.P. Tew, The Journal of Chemical Physics 2018

Group Symmetry

- Abelian group symmetries can be mapped to the cyclic group, which can be used to define a block-sparse form of the tensors (here represented using extra modes), e.g.,

$$w_{aA,bB,iI,jJ} = \sum_{k,K,l,L} u_{aA,bB,kK,lL} v_{kK,lL,iI,jJ}$$

where for some group size G, we have symmetries, e.g.,

$$w_{aA,bB,iI,jJ} \neq 0 \text{ if } A + B - I - J \equiv 0 \pmod{G}$$
$$u_{aA,bB,kK,lL} \neq 0 \text{ if } A + B + K + L \equiv 0 \pmod{G}$$
$$v_{kK,lL,iI,jJ} \neq 0 \text{ if } K + L - I - J \equiv 0 \pmod{G}$$

- We can write each of these tensors using a reduced form and a Kronecker delta tensor,

$$w_{aA,bB,iI,jJ} = r^{(W)}_{aA,bB,iI,j} \delta^{(W)}_{ABIJ}$$

where $\delta^{(W)}_{ABIJ} = 1$ if $A + B - I - J \equiv 0 \pmod{G}$ and $\delta^{(W)}_{ABIJ} = 0$ otherwise

Block Contraction Approach to Group Symmetry

Such symmetries are often handled by indirect indexing in nested loops

Algorithm 2.1 Loop nest to perform group symmetric contraction $w_{aA,bB,iI,jJ} = \sum_{k,K,l,L} u_{aA,bB,kK,lL} v_{kK,lL,iI,jJ}$ using standard reduced forms $\bar{w}_{aA,bB,iI,j}$, $\bar{u}_{aA,bB,kK,l}$, and $\bar{v}_{kK,lL,iI,j}$.

for $A = 1, \ldots, G$ **do**
  for $B = 1, \ldots, G$ **do**
    for $I = 1, \ldots, G$ **do**
      $J = A + B - I \bmod G$
      for $K = 1, \ldots, G$ **do**
        $L = A + B - K \bmod G$
        $\forall a, b, i, j, \quad \bar{w}_{aA,bB,iI,j} = \bar{w}_{aA,bB,iI,j} + \sum_{k,l} \bar{u}_{aA,bB,kK,l} \bar{v}_{kK,lL,iI,j}$
      end for
    end for
  end for
end for

However, transformations of tensors are also possible to reduce such contractions to a "direct product", which has previously been done for group symmetric tensor contractions in quantum chemistry[1,2]

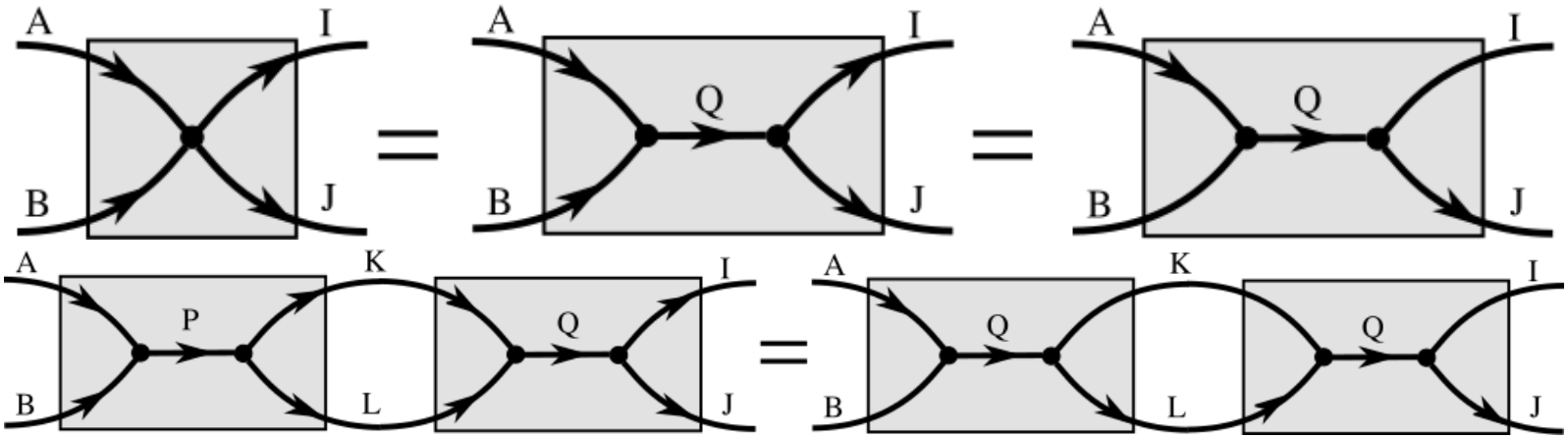
[1] J.F. Stanton, J. Gauss, J.D. Watts, and R.J. Bartlett, The Journal of Chemical Physics 1991

[2] D. Matthews, Molecular Physics 2019

Group Symmetry in Tensor Contractions

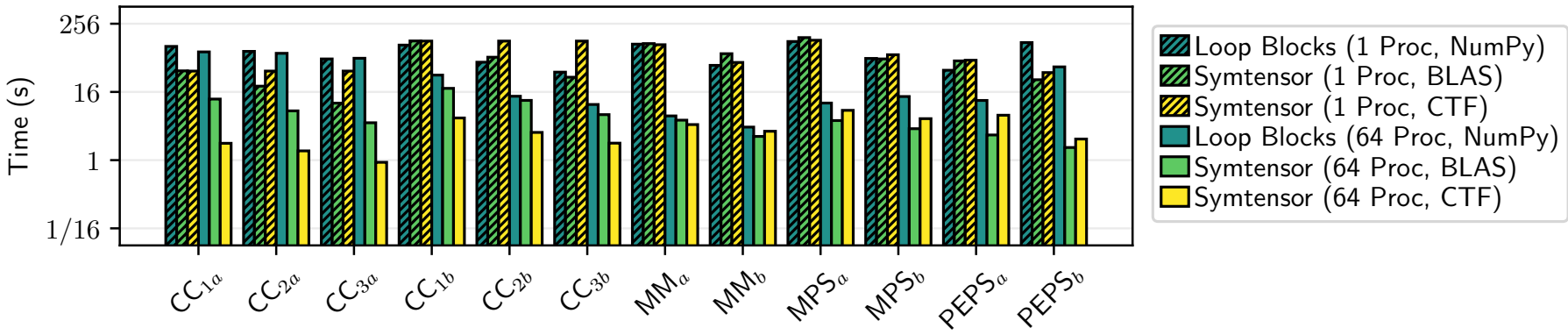
New contraction algorithm, *irreducible representation alignment* uses new reduced form to handle group symmetry (momentum conservation, spin, quantum numbers, etc.) without looping over blocks or sparsity[1]

$$w_{ABIJ} = \sum_{KL} \bar{r}^{(U)}_{ABK} \underbrace{\delta^{(U)}_{ABKL}\delta^{(V)}_{KLIJ}}_{\sum_Q \delta^{(1)}_{ABQ}\delta^{(2)}_{IJQ}\delta^{(3)}_{KLQ}} \bar{r}^{(V)}_{KIJ} = \sum_Q \delta^{(1)}_{ABQ}\delta^{(3)}_{IJQ} \underbrace{\sum_K r^{(U)}_{AKQ} r^{(V)}_{KIQ}}_{r^{(W)}_{AIQ}}$$



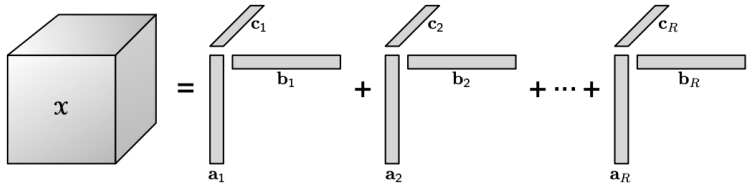
[1] Y. Gao, P. Helms, G. Chan, and E.S., arXiv:2007.08056

Automation of Group Symmetric Contractions



- Group symmetric tensors represented programmatically by
  - a dense reduced tensor (containing unique data)
  - an implicit sparse tensor (Kronecker delta tensor) describing the group symmetry
- At contraction time reduced form are aligned by contraction with Kronecker delta tensor (Q index is introduced)
- Users can write symmetry-oblivious code

CP Decomposition



- For a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n \times n \times n}$, the CP decomposition[1,2] is defined by matrices $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ such that

$$t_{ijk} = \sum_{r=1}^{R} u_{ir} v_{jr} w_{kr}$$

[1] F.L. Hitchcock, Studies in Applied Mathematices 1927

[2] T. Kolda and B. Bader, SIAM Review 2009

CP Decomposition for 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

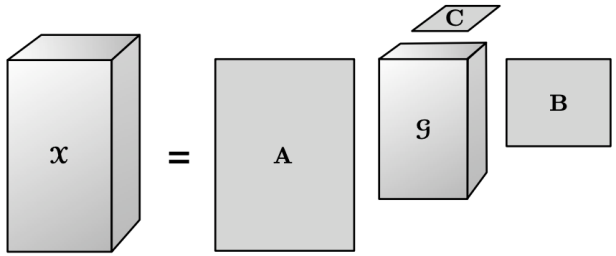
$$(ab|ij) = \sum_p d_{abp} d^*_{ijp}$$

- The tensor hypercontraction 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 this factorization is also applied to the amplitude tensor, CCSD scaling can be theoretically further reduced to $O(n^4)$

Tucker Decomposition



- The Tucker decomposition[1] expresses an order d tensor via a smaller order d core tensor and d factor matrices
  - For a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n \times n \times n}$, the Tucker decomposition is defined by core tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ and factor matrices $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ with orthonormal columns, such that

$$t_{ijk} = \sum_{p=1}^{R_1} \sum_{q=1}^{R_2} \sum_{r=1}^{R_3} z_{pqr} u_{ip} v_{jq} w_{kr}$$

  - If an exact Tucker decomposition exists, it can be computed via SVD (HoSVD)
  - HOOI method optimizes in an alternating manner among $(\boldsymbol{U}, \boldsymbol{\mathcal{Z}})$, $(\boldsymbol{V}, \boldsymbol{\mathcal{Z}})$, $(\boldsymbol{W}, \boldsymbol{\mathcal{Z}})$

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

Recent Work on Tensor Decompositions

Our group has a number of recent developments in algorithms and parallel software for tensor decomposition optimization algorithms

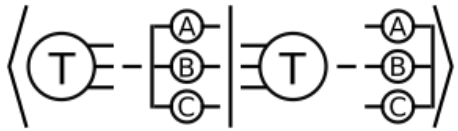
- Navjot Singh, Linjian Ma, Hongru Yang, and ES. *Comparison of accuracy and scalability of Gauss-Newton and alternating least squares for CP decomposition*, arXiv:1910.12331 (SISC 2021).
- Linjian Ma and ES. *Accelerating alternating least squares for tensor decomposition by pairwise perturbation*, arXiv:1811.10573 (NLAA 2022).
- Linjian Ma and ES. *Efficient parallel CP decomposition with pairwise perturbation and multi-sweep dimension tree*, arXiv:2010.12056 (IPDPS 2021).
- Linjian Ma and ES. *Fast and accurate randomized algorithms for low-rank tensor decompositions*, arxiv.org:2104.0110 (NeurIPS 2021).

A Distance Metric for Well-Conditioned CP Decomposition

- CP decomposition algorithms usually minimize the Frobenius norm

$$\|\boldsymbol{\mathcal{T}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 = \|\operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!])\|_2^2$$

$$= \sum_{i,j,k} \left(t_{ijk} - \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}\right)^2$$



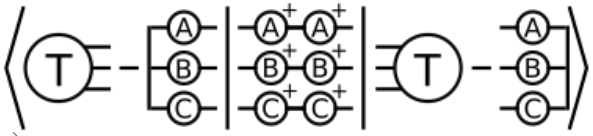
- Ardavan Afshar et al [AAAI 2021] minimize Wasserstein distance, improving robustness for downstream tasks
- We consider Mahalanobis distance based on covariance matrices[1]

$$\|\operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!])\|_{\boldsymbol{M}^{-1}}^2 = \boldsymbol{r}^T \boldsymbol{M}^{-1} \boldsymbol{r}$$

where $\boldsymbol{r} = \operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!])$

and $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{A}^T \otimes \boldsymbol{B}\boldsymbol{B}^T \otimes \boldsymbol{C}\boldsymbol{C}^T$

$$+(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^+) \otimes (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^+) \otimes (\boldsymbol{I} - \boldsymbol{C}\boldsymbol{C}^+)$$

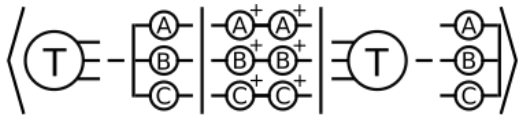


[1] Navjot Singh and E.S., Alternating Mahalanobis Distance Minimization for Stable and Accurate CP Decomposition, arXiv:2204.07208

Alternating Mahalanobis Distance Minimization (AMDM)

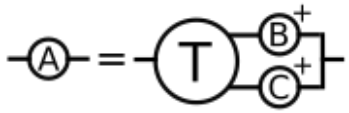
- Optimizing the new metric

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \| \operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}(\llbracket \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \rrbracket) \|^2_{\boldsymbol{M}^{-1}}$$



in an alternating manner yields ALS-like updates

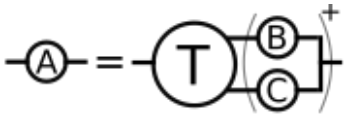
$$\boldsymbol{A} = \boldsymbol{T}_{(1)} (\boldsymbol{C}^{+T} \odot \boldsymbol{B}^{+T})$$



where $\boldsymbol{M}^{+}$ denotes the pseudoinverse of matrix $\boldsymbol{M}$

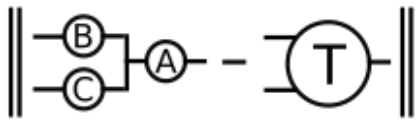
- By comparison, the ALS algorithm computes

$$\boldsymbol{A} = \boldsymbol{T}_{(1)} (\boldsymbol{C} \odot \boldsymbol{B})^{+T}$$



- Both $\boldsymbol{C}^{+T} \odot \boldsymbol{B}^{+T}$ and $(\boldsymbol{C} \odot \boldsymbol{B})^{+T}$ are left inverses of $\boldsymbol{C} \odot \boldsymbol{B}$, suitable for minimizing

$$\min_{\boldsymbol{A}} \| (\boldsymbol{C} \odot \boldsymbol{B}) \boldsymbol{A}^T - \boldsymbol{T}^T_{(1)} \|$$



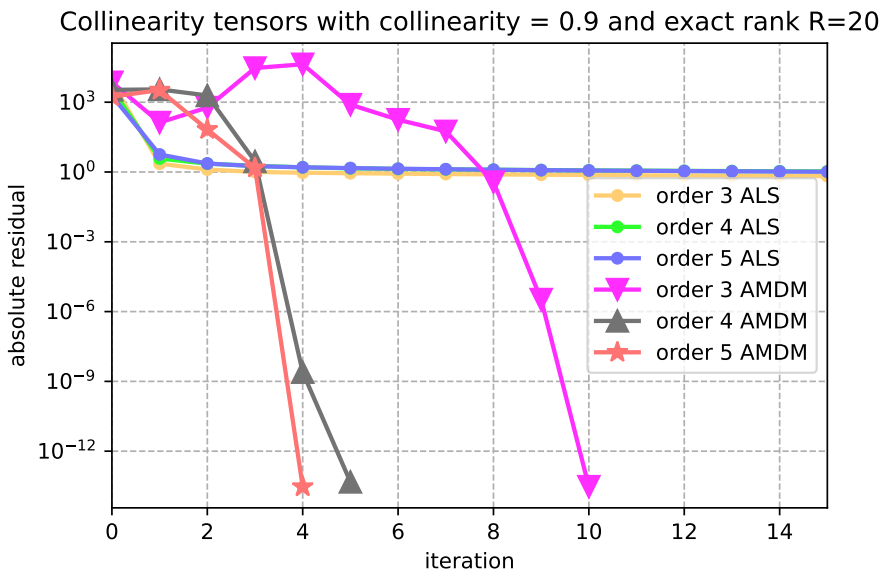
Convergence to Exact Decomposition

When seeking an exact decomposition for a rank $R \leq s$ tensor

- ALS achieves a linear convergence rate[1]
- High-order convergence possible by optimizing all variables via Gauss-Newton,[2,3,4] but is costly per iteration relative to ALS
- AMDM achieves at least quartic order local convergence per sweep of alternating updates
  - error from true solution after solving for one factor scales with product of errors of other factors
- cost per iteration is roughly the same as ALS (dominated by single matricized tensor times Khatri-Rao product (MTTKRP))

[1] A. Uschmajew, SIMAX 2012.
[2] P. Paatero, Chemometrics and Intelligent Laboratory Systems 1997.
[3] A.H. Phan, P. Tichavsky, A. Cichocki, SIMAX 2013.
[4] N. Singh, L. Ma, H. Yang, E.S., SISC 2021.

Exact Decomposition Experimental Performance

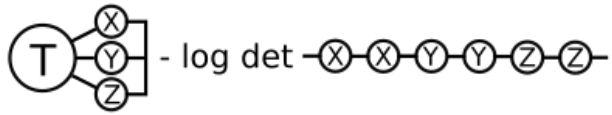


- AMDM achieves high-order convergence for exact decomposition of synthetic random low-rank problems

Properties of Fixed Points of AMDM

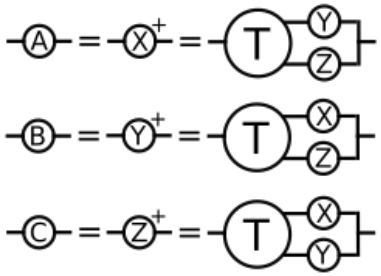
- When $\mathrm{rank}(\boldsymbol{\mathcal{T}}) > R$, consider an AMDM fixed point, $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$
- $\boldsymbol{X} = \boldsymbol{A}^{+T}$, $\boldsymbol{Y} = \boldsymbol{B}^{+T}$, $\boldsymbol{Z} = \boldsymbol{C}^{+T}$ yield a critical point of

$$f(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \langle \boldsymbol{\mathcal{T}}, [\![\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}]\!] \rangle - \log(\det(\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{Z}^T \boldsymbol{Z}))$$



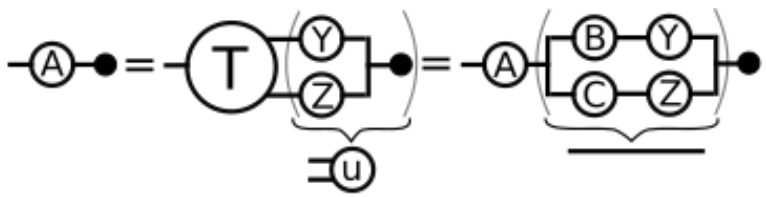
and satisfy tensor-eigenvector-like equations:

$$\boldsymbol{A} = \boldsymbol{X}^{+T} = \boldsymbol{T}_{(1)}(\boldsymbol{Z} \odot \boldsymbol{Y})$$
$$\boldsymbol{B} = \boldsymbol{Y}^{+T} = \boldsymbol{T}_{(2)}(\boldsymbol{Z} \odot \boldsymbol{X})$$
$$\boldsymbol{C} = \boldsymbol{Z}^{+T} = \boldsymbol{T}_{(3)}(\boldsymbol{Y} \odot \boldsymbol{X})$$

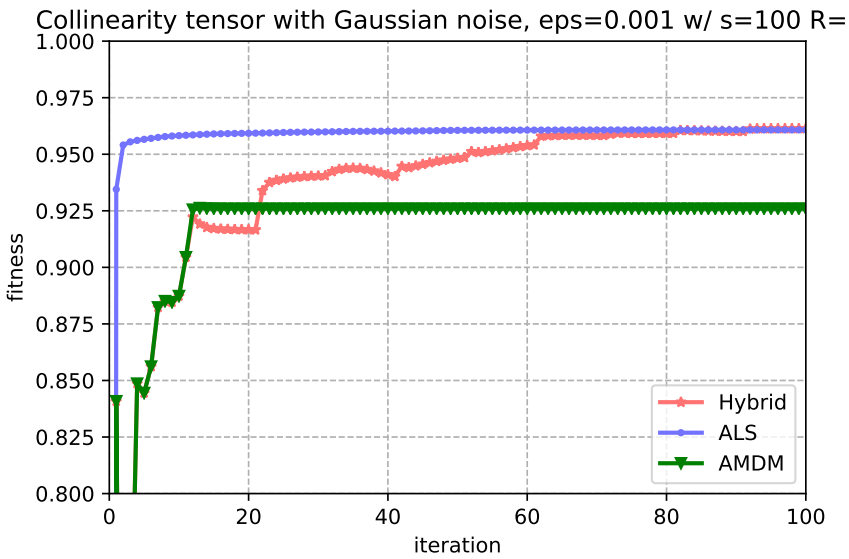


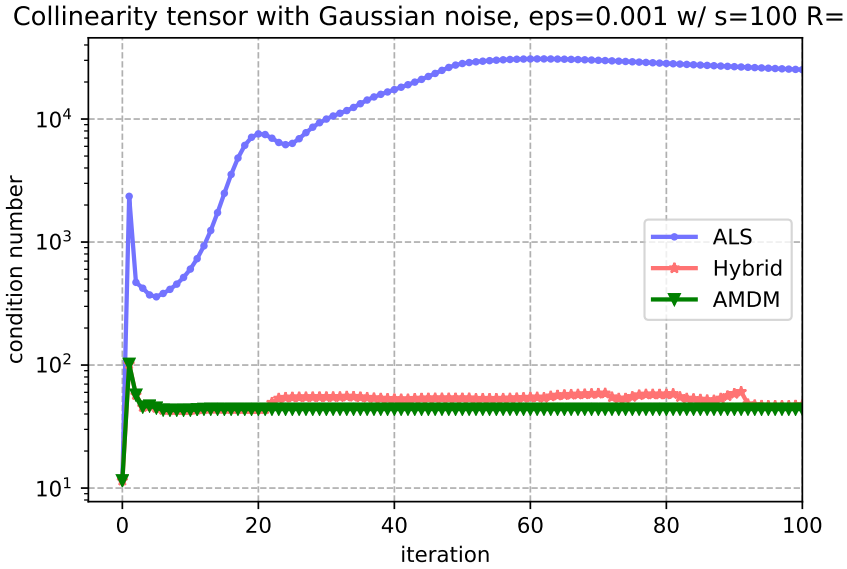
- The reconstructed tensor $\tilde{\boldsymbol{\mathcal{T}}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ exactly represents the action of the original tensor on vectors in the support of the factors

$$\boldsymbol{T}_{(1)}\boldsymbol{u} = \tilde{\boldsymbol{T}}_{(1)}\boldsymbol{u}, \quad \forall \boldsymbol{u} \in \mathrm{span}(\boldsymbol{C} \odot \boldsymbol{B})$$
$$\boldsymbol{T}_{(2)}\boldsymbol{v} = \tilde{\boldsymbol{T}}_{(2)}\boldsymbol{v}, \quad \forall \boldsymbol{v} \in \mathrm{span}(\boldsymbol{C} \odot \boldsymbol{A})$$
$$\boldsymbol{T}_{(3)}\boldsymbol{w} = \tilde{\boldsymbol{T}}_{(3)}\boldsymbol{w}, \quad \forall \boldsymbol{w} \in \mathrm{span}(\boldsymbol{B} \odot \boldsymbol{A})$$



Approximate Decomposition Results with AMDM

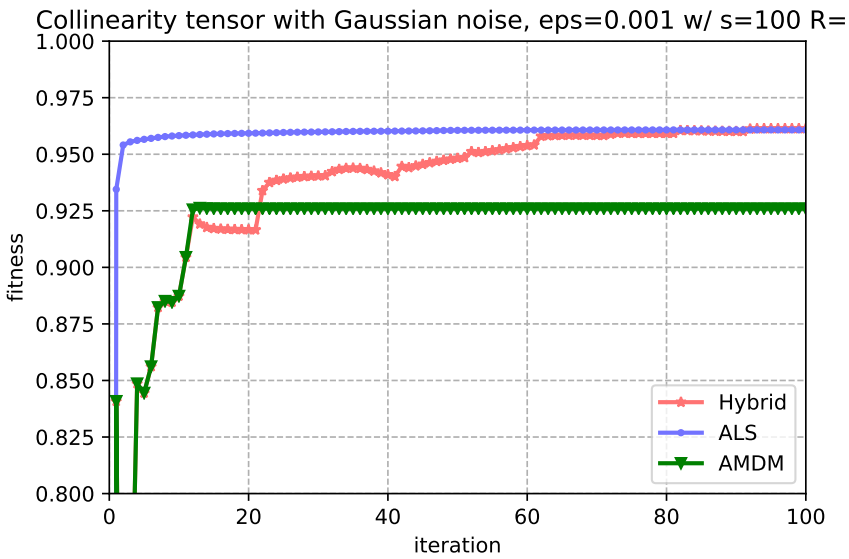


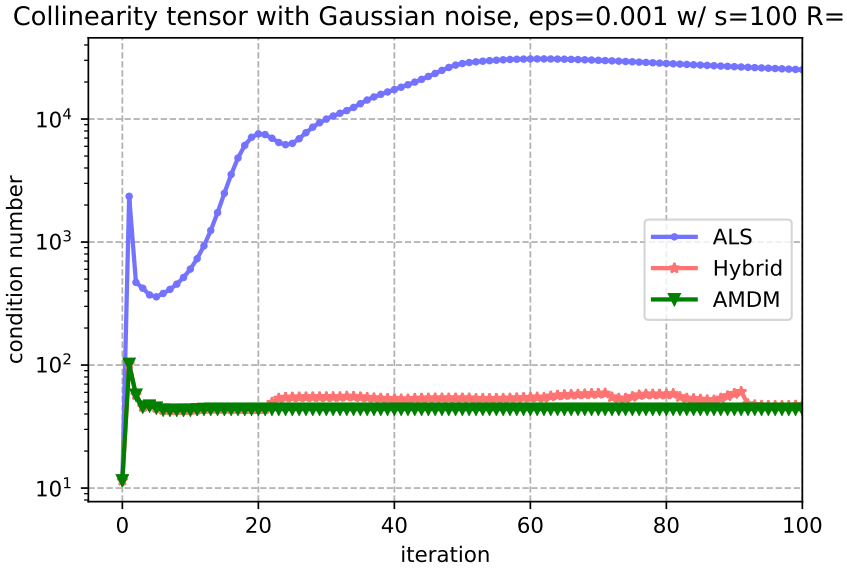


- AMDM finds decomposition with lower CP condition number[1]
- Hybrid version gradually transitions from basic AMDM to ALS

[1] P. Breiding and N. Vannieuwenhoven, SIMAX 2018.

Approximate Decomposition Results with AMDM

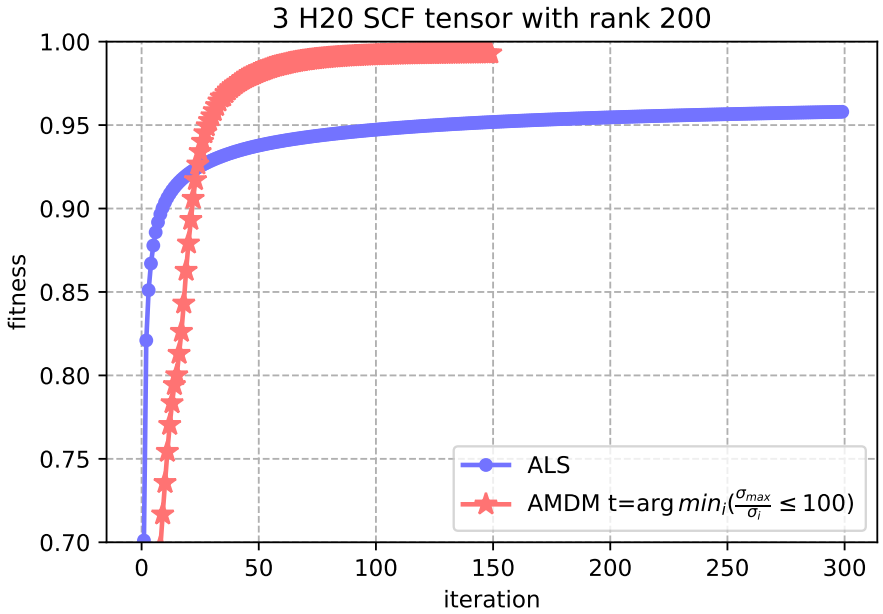




- AMDM finds decomposition with lower CP condition number[1]
- Hybrid version gradually transitions from basic AMDM to ALS

[1] P. Breiding and N. Vannieuwenhoven, SIMAX 2018.

AMDM for Quantum Chemistry Tensors

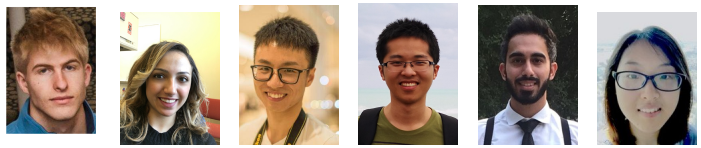


- AMDM can also be applied when CP rank exceeds tensor dimension
- Hybrid version is effective in initial experiments on density fitting intermediate tensors (for tensor hypercontraction construction)

Acknowledgements

- Laboratory for Parallel Numerical Algorithms (LPNA) at University of Illinois
- This work has been support by 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