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
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., \( v \in \mathbb{R}^n \)
- Order 2 tensors are matrices, e.g., \( A \in \mathbb{R}^{m \times n} \)
- An order 3 tensor with dimensions \( s_1 \times s_2 \times s_3 \) is denoted as \( T \in \mathbb{R}^{s_1 \times s_2 \times s_3} \) with elements \( t_{ijk} \) for \( i \in \{1, \ldots, s_1\}, j \in \{1, \ldots, s_2\}, k \in \{1, \ldots, s_3\} \)
A **tensor contraction** describes a set of products and sums of elements from two tensors.

<table>
<thead>
<tr>
<th>Tensor Contraction</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>inner product</td>
<td>$w = \sum_i u_i v_i$</td>
</tr>
<tr>
<td>outer product</td>
<td>$w_{ij} = u_i v_{ij}$</td>
</tr>
<tr>
<td>pointwise product</td>
<td>$w_i = u_i v_i$</td>
</tr>
<tr>
<td>Hadamard product</td>
<td>$w_{ij} = u_{ij} v_{ij}$</td>
</tr>
<tr>
<td>matrix multiplication</td>
<td>$w_{ij} = \sum_k u_{ik} v_{kj}$</td>
</tr>
<tr>
<td>batched mat.-mul.</td>
<td>$w_{ijl} = \sum_k u_{ikl} v_{kjl}$</td>
</tr>
<tr>
<td>tensor times matrix</td>
<td>$w_{ilk} = \sum_j u_{ijk} v_{lj}$</td>
</tr>
</tbody>
</table>

Tensor contractions are prevalent in quantum chemistry methods.
General Tensor Contractions

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

$$w_{i_1...i_sj_1...j_t} = \sum_{k_1...k_v} u_{i_1...i_s k_1...k_v} v_{k_1...k_v j_1...j_t}$$

- 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 $C = AB$ where $C \in \mathbb{R}^{n^s \times n^t}$, $A \in \mathbb{R}^{n^s \times n^v}$, and $B \in \mathbb{R}^{n^v \times n^t}$
- 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

- 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
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,

\[ w_{aA,bB,iI,jJ} = r_{aA,bB,iI,jJ}^{(W)} \delta^{(W)}_{ABIJ} \]

where \( \delta^{(W)}_{ABIJ} = 1 \text{ if } A + B - I - J \equiv 0 \pmod{G} \) 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 \( w_{aA, bB, iI, jJ}, u_{aA, bB, kK, lL} \), and \( v_{kK, lL, iI, jJ} \).

\[
\begin{align*}
&\text{for } A = 1, \ldots, G \text{ do} \\
&\quad \text{for } B = 1, \ldots, G \text{ do} \\
&\quad \quad \text{for } I = 1, \ldots, G \text{ do} \\
&\quad \quad \quad J = A + B - I \mod G \\
&\quad \quad \quad \text{for } K = 1, \ldots, G \text{ do} \\
&\quad \quad \quad \quad L = A + B - K \mod G \\
&\quad \quad \quad \forall a, b, i, j, \quad w_{aA, bB, iI, jJ} = w_{aA, bB, iI, jJ} + \sum_{k, l} u_{aA, bB, kK, lL} v_{kK, lL, iI, jJ} \\
&\quad \quad \end{align*}
\]

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

\[
\begin{align*}
    w_{ABIJ} &= \sum_{KL} \bar{r}^{(U)}_{ABK} \delta^{(U)}_{ABKL} \delta^{(V)}_{KLIJ} \bar{r}^{(V)}_{KIJ} \\
    &= \sum_{Q} \delta^{(1)}_{ABQ} \delta^{(3)}_{IJQ} \sum_{K} \delta^{(1)}_{AKQ} \delta^{(3)}_{IJQ} \sum_{K} \bar{r}^{(U)}_{AKQ} \bar{r}^{(V)}_{KIQ}
\end{align*}
\]

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 $\mathbf{T} \in \mathbb{R}^{n \times n \times n}$, the CP decomposition\(^1,2\) is defined by matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ such that

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

---

1. F.L. Hitchcock, Studies in Applied Mathematics 1927
2. T. Kolda and B. Bader, SIAM Review 2009
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 $X$

When this factorization is also applied to the amplitude tensor, CCSD scaling can be theoretically further reduced to $O(n^4)$
The **Tucker decomposition**\(^1\) expresses an order \(d\) tensor via a smaller order \(d\) core tensor and \(d\) factor matrices.

For a tensor \(\mathbf{T} \in \mathbb{R}^{n \times n \times n}\), the Tucker decomposition is defined by core tensor \(\mathbf{Z} \in \mathbb{R}^{R_1 \times R_2 \times R_3}\) and factor matrices \(\mathbf{U}, \mathbf{V},\) and \(\mathbf{W}\) 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 \((\mathbf{U}, \mathbf{Z}), (\mathbf{V}, \mathbf{Z}), (\mathbf{W}, \mathbf{Z})\)

---

\(^1\) 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


A Distance Metric for Well-Conditioned CP Decomposition

- CP decomposition algorithms usually minimize the Frobenius norm

\[
\|T - [A, B, C]\|_F^2 = \|\text{vec}(T) - \text{vec}([A, B, C])\|_2^2
\]

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

- Ardavan Afshar et al [AAAI 2021] minimize Wasserstein distance, improving robustness for downstream tasks

- We consider Mahalanobis distance based on covariance matrices\(^1\)

\[
\|\text{vec}(T) - \text{vec}([A, B, C])\|_{M^{-1}}^2 = r^T M^{-1} r
\]

where \( r = \text{vec}(T) - \text{vec}([A, B, C]) \)

and \( M = AA^T \otimes BB^T \otimes CC^T \)

\[
+ (I - AA^+) \otimes (I - BB^+) \otimes (I - CC^+)
\]

\(^1\)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

\[
\min_{A,B,C} \| \text{vec}(T) - \text{vec}([A, B, C]) \|^2_{M^{-1}}
\]

in an alternating manner yields ALS-like updates

\[
A = T_{(1)}(C^+T \odot B^+T)
\]

where \(M^+\) denotes the pseudoinverse of matrix \(M\)

- By comparison, the ALS algorithm computes

\[
A = T_{(1)}(C \odot B)^+T
\]

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

\[
\min_A \|(C \odot B)A^T - T_{(1)}^T\|
\]
When seeking an exact decomposition for a rank \( R \leq s \) tensor

- ALS achieves a \textit{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 \textit{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
- \textit{cost} per iteration is roughly the \textit{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.
AMDM achieves high-order convergence for exact decomposition of synthetic random low-rank problems
Properties of Fixed Points of AMDM

- When \( \text{rank}(\mathcal{T}) > R \), consider an AMDM fixed point, \( A, B, C \)
- \( X = A^{+T}, \ Y = B^{+T}, \ Z = C^{+T} \) yield a critical point of
  \[
  f(X, Y, Z) = \langle \mathcal{T}, [X, Y, Z] \rangle - \log(\det(X^TXY^TYZ^TZ))
  \]
  and satisfy tensor-eigenvector-like equations:
  \[
  A = X^{+T} = T_1(Z \otimes Y)
  B = Y^{+T} = T_2(Z \otimes X)
  C = Z^{+T} = T_3(Y \otimes X)
  \]
- The reconstructed tensor \( \tilde{\mathcal{T}} = [A, B, C] \) exactly represents the action of the original tensor on vectors in the support of the factors
  \[
  T_1u = \tilde{T}_1u, \quad \forall u \in \text{span}(C \otimes B)
  T_2v = \tilde{T}_2v, \quad \forall v \in \text{span}(C \otimes A)
  T_3w = \tilde{T}_3w, \quad \forall w \in \text{span}(B \otimes A)
  \]
• 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 finds decomposition with lower CP condition number\(^\dagger\)

Hybrid version gradually transitions from basic AMDM to ALS

\(^\dagger\)P. Breiding and N. Vannieuwenhoven, SIMAX 2018.
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 supported by NSF awards #1839204 (RAISE-TAQS), #1931258 (CSSI), #1942995 (CAREER)
- Stampede2 resources at TACC via XSEDE

http://lpna.cs.illinois.edu