Scalable Algorithms for Tensor Computations

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

SIAM PP, Seattle WA
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
Outline

1. Introduction
2. Matrix Factorizations
3. Tensor Computations
4. Software Libraries
5. Conclusion
3D algorithms for matrix computations

For Cholesky factorization with $p$ processors, BSP (critical path) costs are

\[ F = \Theta(n^3/p), \quad W = \Theta(n^2/\sqrt{cp}), \quad S = \Theta(\sqrt{cp}) \]

using $c$ matrix copies (processor grid is 2D for $c = 1$, 3D for $c = p^{1/3}$).

Achieving similar costs for LU, QR, and the symmetric eigenvalue problem requires algorithmic changes.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Square TRSM ✓</th>
<th>Rectangular TRSM ✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangular solve</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>LU with pivoting</td>
<td>pairwise pivoting ✓</td>
<td>tournament pivoting ✓</td>
</tr>
<tr>
<td>QR factorization</td>
<td>Givens on square ✓</td>
<td>Householder on rect. ✓</td>
</tr>
<tr>
<td>Sym. eig.</td>
<td>eigenvalues only ✓</td>
<td>eigenvectors X</td>
</tr>
</tbody>
</table>

✓ means costs attained (synchronization within polylog factors).

---

1. B. Lipshitz, MS thesis 2013
2. T. Wicky, E.S., T. Hoefler, IPDPS 2017
3. A. Tiskin, FGCS 2007
4. E.S., J. Demmel, EuroPar 2011
5. E.S., G. Ballard, T. Hoefler, J. Demmel, SPAA 2017
New algorithms can circumvent lower bounds

For TRSM, we can achieve a lower synchronization/communication cost by performing **triangular inversion on diagonal blocks**

- decreases synchronization cost by $O(p^{2/3})$ on $p$ processors with respect to known algorithms
- optimal communication for **any number of right-hand sides**
- MS thesis work by Tobias Wicky$^1$

---

$^1$ T. Wicky, E.S., T. Hoefler, IPDPS 2017
Cholesky-QR2 for rectangular matrices

Cholesky-QR2\(^1\) with 3D Cholesky gives a practical 3D QR algorithm\(^2\)

- Compute \(A = \hat{Q}R\) using Cholesky \(A^TA = R^TR\)
- Correct computed factorization by Cholesky-QR of \(\hat{Q}\)
- Attains full accuracy so long as \(\text{cond}(A) < 1/\sqrt{\epsilon_{\text{mach}}}\)

---

1. T. Fukaya, Y. Nakatsukasa, Y. Yanagisawa, Y. Yamamoto, 2014
2. E. Hutter, E.S., IPDPS 2019
Reducing the symmetric matrix $A \in \mathbb{R}^{n \times n}$ to a tridiagonal matrix 

$$T = Q^T AQ$$

by an orthogonal similarity transformation is most costly in eigenvalue computation (SVD is similar).

- can be done by successive column QR factorizations

$$T = \underbrace{Q_n^T \cdots Q_1^T}_{Q^T} A \underbrace{Q_1 \cdots Q_n}_{Q}$$

- two-sided updates harder to manage than one-sided
- can use $n/b$ QRs on panels of $b$ columns to go to band-width $b + 1$
- $b = 1$ gives direct tridiagonalization
Writing the orthogonal transformation in Householder form, we get

\[
(I - UTU^T)A(I - UTU^T) = A - UV^T - VU^T
\]

where columns of \( U \) contains Householder vectors and \( V \) is

\[
V^T = TU^T + \frac{1}{2} T^T U^T A U T U^T
\]

- if \( b = 1 \), \( U \) is a column-vector, and \( AU \) is dominated by vertical communication cost (moving \( A \) between memory and cache)
- idea: reduce to banded matrix \((b \gg 1)\) first\(^1\)

\(^1\) T. Auckenthaler, H. Bungartz, T. Huckle, L. Krämer, B. Lang, P. Willems 2011
Successive band reduction (SBR)

After reducing to a banded matrix, we need to transform the banded matrix to a tridiagonal one

- fewer nonzeros lead to lower computational cost, \( F = O(n^2b/p) \)
- however, transformations introduce fill/bulges
- bulges must be chased down the band\(^1\)

communication- and synchronization-efficient 1D SBR algorithm known for small band-width\(^2\)

---

1 B. Lang 1993; C.H. Bischof, B. Lang, X. Sun 2000
2 G. Ballard, J. Demmel, N. Knight 2012
Communication-efficient eigenvalue computation

Previous work (start-of-the-art): two-stage tridiagonalization
- implemented in ELPA, can outperform ScaLAPACK\(^1\)
- with \( n = n / \sqrt{p} \), 1D SBR gives \( W = O(n^2 / \sqrt{p}) \), \( S = O(\sqrt{p} \log^2(p)) \)

New results\(^3\): many-stage tridiagonalization
- \( \Theta(\log(p)) \) intermediate band-widths to achieve \( W = O(n^2 / p^{2/3}) \)
- communication-efficient rectangular QR with processor groups

3D SBR (each QR and matrix multiplication update parallelized)

---

\(^1\) T. Auckenthaler, H. Bungartz, T. Huckle, L. Krämer, B. Lang, P. Willems 2011
\(^2\) G. Ballard, J. Demmel, N. Knight 2012
\(^3\) E.S., G. Ballard, J. Demmel, T. Hoefler 2017
Symmetric eigen solver results summary

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$W$</th>
<th>$Q$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScalAPACK</td>
<td>$n^2/\sqrt{p}$</td>
<td>$n^3/p$</td>
<td>$n \log(p)$</td>
</tr>
<tr>
<td>ELPA</td>
<td>$n^2/\sqrt{p}$</td>
<td></td>
<td>$n \log(p)$</td>
</tr>
<tr>
<td>two-stage + 1D-SBR</td>
<td>$n^2/\sqrt{p}$</td>
<td>$n^2 \log(n)/\sqrt{p}$</td>
<td>$\sqrt{p}(\log^2(p) + \log(n))$</td>
</tr>
<tr>
<td>many-stage</td>
<td>$n^2/p^{2/3}$</td>
<td>$n^2 \log(p)/p^{2/3}$</td>
<td>$p^{2/3} \log^2 p$</td>
</tr>
</tbody>
</table>

- costs are asymptotic (same computational cost $F$ for eigenvalues)
- $W$ – horizontal (interprocessor) communication
- $Q$ – vertical (memory–cache) communication excluding $W + F/\sqrt{H}$ where $H$ is cache size
- $S$ – synchronization cost (number of supersteps)
Tensor Decompositions

- Tensor of order $N$ has $N$ modes and dimensions $s \times \cdots \times s$
- Canonical polyadic (CP) tensor decomposition\footnote{T.G. Kolda and B.W. Bader, SIAM Review 2009}

\[
x = a_1 c_1 b_1 + a_2 c_2 b_2 + \cdots + a_N c_N b_N
\]

- 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
New algorithm: pairwise perturbation (PP)\(^1\) approximates ALS

- accurate when ALS sweep over factor matrices stagnates
- rank \( R < s^{N-1} \) CP decomposition:
  - ALS sweep cost \( O(s^N R) \Rightarrow O(s^2 R) \), up to 600x speed-up
- rank \( R < s \) Tucker decomposition:
  - ALS sweep cost \( O(s^N R) \Rightarrow O(s^2 R^{N-1}) \)

\(^1\) L. Ma, E.S. arXiv:1811.10573

Linjian Ma
New regularization scheme\textsuperscript{1} for Gauss-Newton CP with implicit CG\textsuperscript{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

---

\textsuperscript{1}Navjot Singh, Linjian Ma, Hongru Yang, and E.S. arXiv:1910.12331

\textsuperscript{2}P. Tichavsky, A. H. Phan, and A. Cichocki., 2013
Symmetry in Tensor Contractions

New algebraic transformations for contractions to exploit permutational symmetry\(^1\) and group symmetry\(^2\), prevalent in quantum chemistry/physics

\[
C = AB + BA \Rightarrow
\]

\[
c_{ij} = \sum_k [(a_{ij} + a_{ik} + a_{jk}) \cdot (b_{ij} + b_{ik} + b_{jk})] - \ldots
\]

\(^1\) E.S, J. Demmel, CMAM 2020
\(^2\) collaboration with Yang Gao, Phillip Helms, and Garnet Chan at Caltech
Cyclops Tensor Framework\textsuperscript{1} sparse/dense generalized tensor algebra

- Cyclops is a C++ library that distributes each tensor over MPI
- Used in chemistry (PySCF, QChem)\textsuperscript{2}, quantum circuit simulation (IBM/LLNL)\textsuperscript{3}, and graph analysis (betweenness centrality)\textsuperscript{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

\textsuperscript{1}https://github.com/cyclops-community/ctf
\textsuperscript{2}E.S., D. Matthews, J. Hammond, J.F. Stanton, J. Demmel, JPDC 2014
\textsuperscript{3}E. Pednault, J.A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. S., E. Draeger, E. Holland, and R. Wisnieff, 2017
\textsuperscript{4}E.S., M. Besta, F. Vella, T. Hoefler, SC 2017
Cyclops supports sparse representation of tensors\(^1\)

Choice of representation specified in tensor constructor

CSR or DCSR\(^2\) (2-index CSF\(^3\)) representation used locally for contractions

---

\(^1\) E.S., T. Hoefler 2015
\(^2\) A. Bulúc, J.R. Gilbert, 2008
\(^3\) S. Smith, G. Karypis 2015
All-at-Once Multi-Tensor Contraction

With sparsity, all-at-once contraction of multiple tensors can be faster.

- Sparse tensor CP decomposition is dominated by MTTKRP
- All-at-once sparse MTTKRP needs less communication than pairwise

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

- Tensor times tensor product (TTTP) enables sparse residual and CP tensor completion
- Cost and memory footprint reduced asymptotically

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

1S. Smith, J. Park, G. Karypis, 2018
2Zecheng Zhang, Xiaoxiao Wu, Naijing Zhang, Siyuan Zhang, and E.S. arXiv:1910.02371
Conclusion

- Communication avoiding matrix factorizations
  - 3D algorithms move $O(p^{1/6})$ less data on $p$ processors
  - Faster algorithms for TRSM, Cholesky, LU, QR, and symmetric eigensolve

- Optimization methods for tensor decomposition
  - Pairwise perturbation approximates ALS sweeps with asymptotically less cost
  - Gauss-Newton methods with new regularization scheme more effective than ALS

- Algorithms and software for sparse/symmetric tensors
  - Algebraic reorganizations to exploit permutational and group symmetry
  - Cyclops library for dense/sparse generalized distributed tensor algebra in C++/Python
Acknowledgements

- Laboratory for Parallel Numerical Algorithms (LPNA) at University of Illinois
- Key external collaborators: James Demmel, Torsten Hoefler, Garnet Chan
- NSF OAC CSSI funding (award #1931258)
- DOE Computational Science Graduate Fellowship
- Stampede2 resources at TACC via XSEDE

http://lpna.cs.illinois.edu