

Massively Parallel Polar Decomposition on Distributed-Memory Systems

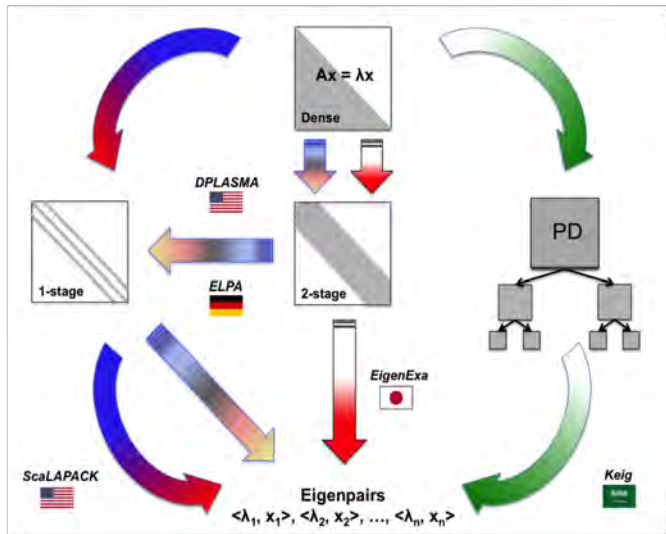
PhD Student Dalal Sukkari

Advisors: Prof. David Keyes, Dr. Hatem Ltaief

Extreme Computing Research Center
King Abdullah University of Science and Technology



The Big Picture (Similar w/ SVD)



Cray LibSci 17.11.1

What is The Polar Decomposition ?

- The polar decomposition:

$$\mathbf{A} = \mathbf{U}_p \mathbf{H}, \quad A \in \mathbb{R}^{m \times n} (m \geq n),$$

where U_p is an orthogonal matrix and $H = \sqrt{A^T A}$ is a symmetric positive semidefinite matrix

- The polar decomposition is a critical numerical algorithm for various applications, including aerospace computations, chemistry, factor analysis

Application to Symmetric Eigensolvers and SVD

The polar decomposition can be used as pre-processing step toward calculating the:

- The SVD $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$:

$$A = U_p H = U_p (V \Sigma V^\top) = (U_p V) \Sigma V^\top = U \Sigma V^\top$$

- The EVD: $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top$, $V = [V_1 V_2]$

$$A = U_p H$$

$$U_p + I = [V_1 \ V_2] \begin{bmatrix} I_k & \mathbf{0} \\ \mathbf{0} & -I_{n-k} \end{bmatrix} [V_1 \ V_2]^* + I$$

$$= [V_1 \ V_2] \begin{bmatrix} I_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [V_1 \ V_2]^*$$

$$= 2V_1 V_1^*$$

$$A_1 = V_1^\top A V_1, A_2 = V_2^\top A V_2$$

Background

-
- 1877 Zolotarev, best rational approximant for the scalar sign function.
- 1902 Autonne, *SMF*, definition of the polar decomposition.
- 1994 Higham and Papadimitriou, *SIAM*, matrix inversion QDWH, *shared-memory* systems.
- 2010 Nakatsukasa et. al, *SIAM*, inverse-free QDWH, theoretical accuracy study.
- 2013 Nakatsukasa and Higham, *SIAM*, QDWH-EIG, QDWH-SVD, theoretical accuracy study.
- 2014 Nakatsukasa, *SIAM*, ZOLO-PD, ZOLO-SVD, ZOLO-EIG, theoretical accuracy study.
- 2016 Sukkari, Ltaief and Keyes, *TOMS*, QDWH-SVD, *block algorithm*, *shared-memory* system equipped with multiple GPUs.
- 2016 Sukkari, Ltaief and Keyes, *Euro-Par*, QDWH, QDWH-SVD, *block algorithm*, *distributed-memory* system.
- 2017 Sukkari, Ltaief, Faverge and Keyes, *TPDS*, QDWH, task-based, *shared-memory* system equipped with multiple GPUs.
- 2018 Sukkari, Ltaief, Esposito, Nakatsukasa and Keyes, *TOPC*, ZOLO-PD, *block algorithm*, *distributed-memory* system.

The Polar Decomposition Algorithms

- QDWH:

$$\begin{bmatrix} \sqrt{c_k} X_k \\ I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,$$

$$X_{k+1} = \frac{b_k}{c_k} X_k + \frac{1}{\sqrt{c_k}} \left(a_k - \frac{b_k}{c_k} \right) Q_1 Q_2^*.$$

- Zolo-PD:

$$\begin{bmatrix} X_k \\ \sqrt{c_{2j-1}} I \end{bmatrix} = \begin{bmatrix} Q_{j1} \\ Q_{j2} \end{bmatrix} R_j,$$

$$X_{k+1} = X_k + \sum_{j=1}^r \frac{a_j}{\sqrt{c_{2j-1}}} Q_{j1} Q_{j2}^*.$$

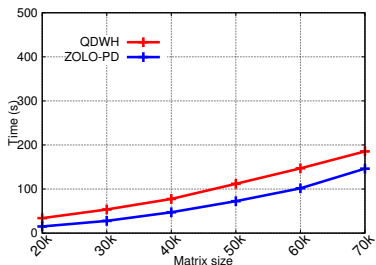
Cray LibSci 17.11.1

Under Consideration by Cray LibSci

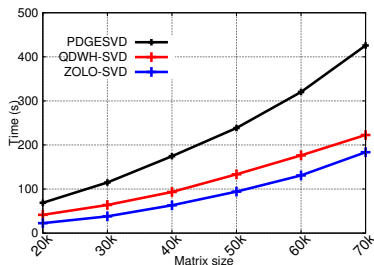
For ill-conditioned matrices, in double precision, QDWH converges after 6 **successive** iterations, while ZOLO-PD converges after 2 **successive** iterations, each execute 8 **independent** embarrassingly parallel factorizations

Arithmetic Complexity and Performance

	QDWH	Successive ZOLOPD	Independent ZOLOPD
# QR-based iter	2	8	1
# Cholesky-based iter	4	8	1
Algorithmic complexity	$33n^3$	$100n^3$	$15n^3$
Memory footprint	$6n^2$	$6n^2$	$48n^2$



(a) Polar Decomposition



(b) SVD solvers

Cray XC40, 800 nodes, Intel Haswell 2.3GHz two-sockets 16 cores

Thank you



جامعة الملك عبد الله
للعلوم والتقنية
King Abdullah University of
Science and Technology

MASSIVELY PARALLEL POLAR DECOMPOSITION ON DISTRIBUTED-MEMORY SYSTEMS

DALAL SUKKARI, HATEM LTAIEF, ANIELLO ESPOSITO, YUJI NAKATSUKASA AND DAVID KEYS

EXTREME COMPUTING RESEARCH CENTER
KING ABDULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

MOTIVATIONS

In order to take advantage of ever-increasing single node core counts for maximizing performance, we investigate the distributed-memory implementation of the polar decomposition (PD) for dense matrices, a fundamental matrix decomposition revealing the nearest orthogonal matrix, and more recently used as the building block for spectral divide-and-conquer algorithms to compute the singular value decomposition (SVD) of a general nonsymmetric matrix as well as the eigenvalue decomposition (EVD) for Hermitian matrices. The current PD employs the univariate, iterative QR-based Dynamically Weighted Halley (QDWH) algorithm. Building upon on QDWH iteration, the key idea lies in finding the best rational approximation for the scalar sign function, which also corresponds to the polar factor for symmetric matrices, to further accelerate the QDWH convergence. Based on the Zolotarev rational functions—introduced by Zolotarev (ZOL) in 1877—this new PD algorithm ZOLO-PD converges within two iterations even for ill-conditioned matrices.

BACKGROUND

The polar decomposition of the matrix $A \in \mathbb{R}^{m \times n}$ ($m \geq n$) is written as $A = U_0 H$, where U_0 is an orthogonal matrix and $H = \sqrt{A^T A}$ is a symmetric positive semidefinite matrix. The matrix sign decomposition and the polar decomposition are the same decomposition when A is Hermitian.

Sign decomposition: $A = SH$
 $S^2 = I, S_+ S_- = I, S_+ S_- S_+ = 0$

Polar decomposition: $A = UH$
 $S^2 H = I, S_+ S_- H = I, S_+ S_- H = 0$

Zolotarev functions:
 $Z_{p,q}(x) = A_n H_{p,q}^{(n)}$
 is the rational approximation to the sign function on $[-1, -\epsilon_1] \cup [1, \epsilon_1]$, where $\epsilon_1 = 1/\sqrt{\kappa(A)}$

QDWH AND ZOLO

QDWH Algorithm $Z_{n,n+1}(t, 0, 0, r + \epsilon)$

- The QDWH iteration can be computed using the mathematically equivalent but numerically more stable QR-based implementation:

$$\begin{bmatrix} \sqrt{r} & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} Q \\ R \end{bmatrix} H;$$

$$X_{i+1} = \frac{1}{r} X_i + \frac{1}{\sqrt{r}} \left(m - \frac{1}{r} \right) Q_i Q_i^T$$

- After a few iterations, the QR-based iteration can be replaced with a *leave-one-out* Cholesky-based iteration.
- Converges in at most six iterations, or DP for matrices with $\kappa(A) < 10^{13}$.



Zolo-PD Algorithm

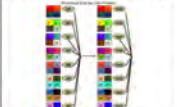
$Z_{p,q}(x) = \frac{U_0 H_{p,q}(x)}{H_{p,q}(x)}$

- Similar to QDWH, $Z_{p,q}(x)$ can be computed stably using QR-based iteration, which can be replaced with Cholesky-based iteration.
- This is ϵ embarrassingly parallel iterations.
- The convergence is attained in just two steps.

$$\begin{bmatrix} X \\ \sqrt{r} & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} Q \\ R \end{bmatrix} H;$$

$$Z_{p,q}(x) = \frac{U_0 H_{p,q}(x)}{H_{p,q}(x)}$$

- This is ϵ embarrassingly parallel iterations.
- The convergence is attained in just two steps.



COMPARISON

	QDWH	Non-parallel	Unconditionally
QR-based iter.	Yes	No	Yes
Mathematically equivalent	Yes	Yes	Yes
Stability	Yes	No	Yes
Memory footprint	Yes	No	Yes

ENVIRONMENT SETTINGS

Software:

- Intel Compiler Suite v13.0.2.164 and v17.0.4.165 on Silicon-Zand Crystal, respectively.
- Cray LibSci/13.2.0/ScalAPACK library.
- MPICH library.

Hardware:

	Crystal	Crystal
V Nodes	4176	4176
Nodes/Node	24	24
Process/Node	24	24
Peak Performance	2.11THz	2.11THz
Peak Performance	4.44THz	4.44THz
Memory Size	128GB	128GB

FUTURE WORK

- Reduce the sparse-matrix overhead for ZOLO-PD at 10^3 nodes / 10^3 iterations / 10^3 iterations.
- Integrate ZOLO-PD into LAPACK library.
- Use ZOLO-PD as building block for SVD and EVD.
- Implement a task-based ZOLO-PD.

REFERENCES

- Y. Nakatsukasa and N. Higham: Stable and Efficient Operator Splitting and Conjugate Algorithms for the Symmetric Eigenvalue Decomposition and the SVD. *SIAM Journal on Scientific Computing*, 35(4):A225–A249, 2013.
- Y. Nakatsukasa and Balesar, Y. Freund: Computing Fundamental Matrix Decompositions Locally via the Matrix Sign Function in Two Iterations. *The Journal of Supercomputing*, 33(4):461–481, 2015.
- H. Ltaief, D. Sukić, A. Esmaili, S. S. Saad, D. K. Sorensen and D. Keys: Memory Efficient Polar Decomposition on Distributed-Memory Systems. *International Journal of Supercomputing*, 27(1):1–12, 2014.
- Y. Nakatsukasa, D. Sukić, S. S. Saad, D. K. Sorensen and D. Keys: Memory Efficient Polar Decomposition on Distributed-Memory Systems. *International Journal of Supercomputing*, 27(1):1–12, 2014.

PERFORMANCE RESULTS

