

Introduction

Target problem: QR factorization of a tall-skinny matrix

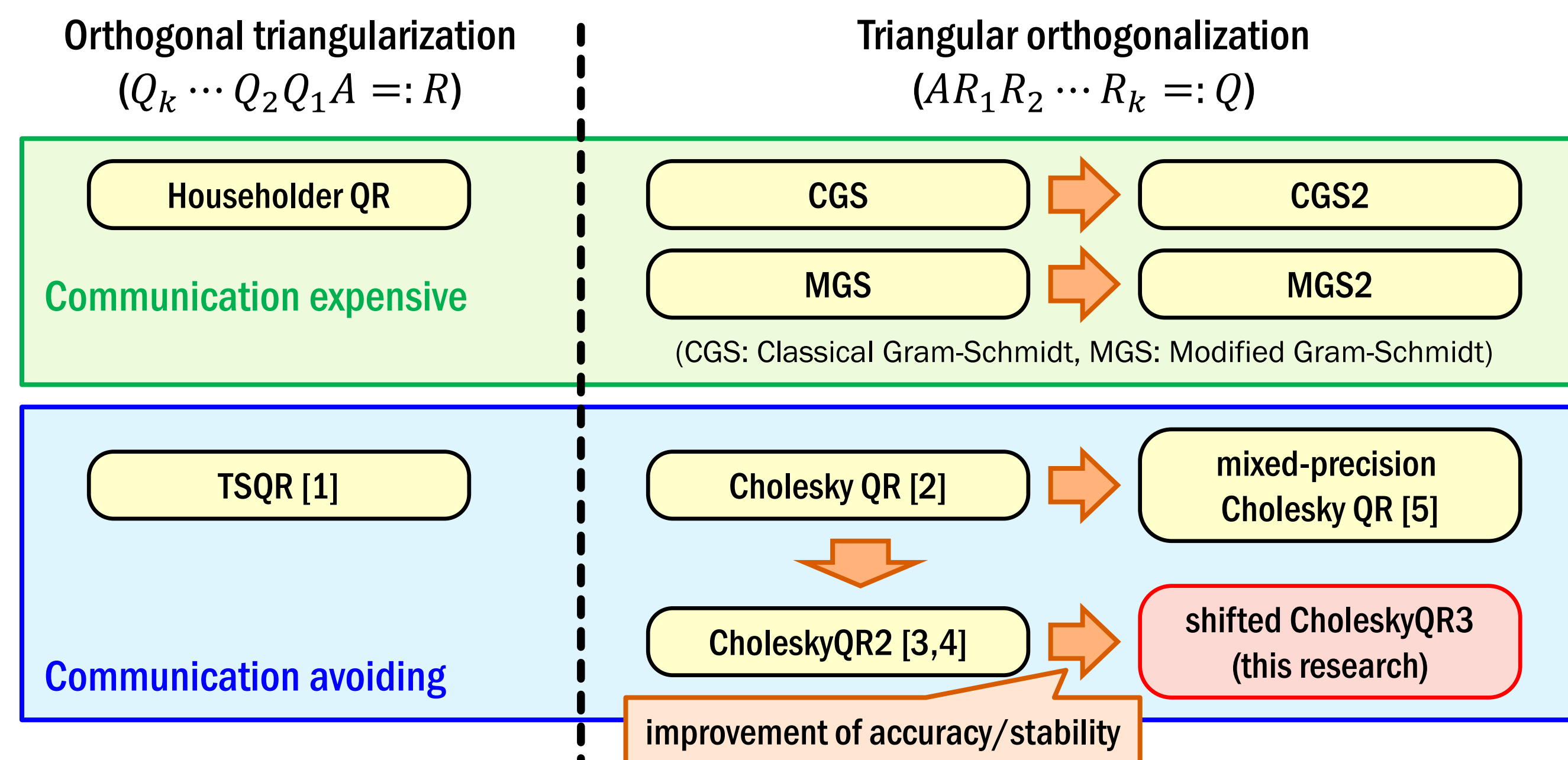
$$A = QR$$

- $A: m \times n$ real and full column rank (mainly focus on $m \gg n$)
- $Q: m \times n$ orthogonal
- $R: n \times n$ upper triangular
- $\kappa_2(A)$: 2-norm condition number ($\kappa_2(A) = \frac{\text{maximum singular value of } A}{\text{minimum singular value of } A}$)

Applications:

- Solution for least-square problems
- Preprocessing of the SVD computation of rectangular matrices
- Orthogonal basis calculation in block projection methods
- Reduced order modeling and structural analysis

The landscape of QR factorization algorithms



The Cholesky QR and CholeskyQR2 algorithms

- Cholesky QR is suitable for high-performance computing but has numerical instability [2].
- CholeskyQR2 has improved stability but usually breaks down when roughly $\kappa_2(A) > 10^8$ [3,4].

CholQR(A) = [Q, R]

- $W := A^T A$
- $R := \text{chol}(W) // \text{Cholesky fact.}$
- $Q := AR^{-1}$

The Cholesky QR algorithm

CholQR2(A) = [Q, R]

- $[A', R_1] := \text{CholQR}(A)$
- $[Q, R_2] := \text{CholQR}(A')$
- $R := R_2 R_1$

The CholeskyQR2 algorithm

The shifted Cholesky QR algorithm

An idea from the viewpoint of triangular orthogonalization

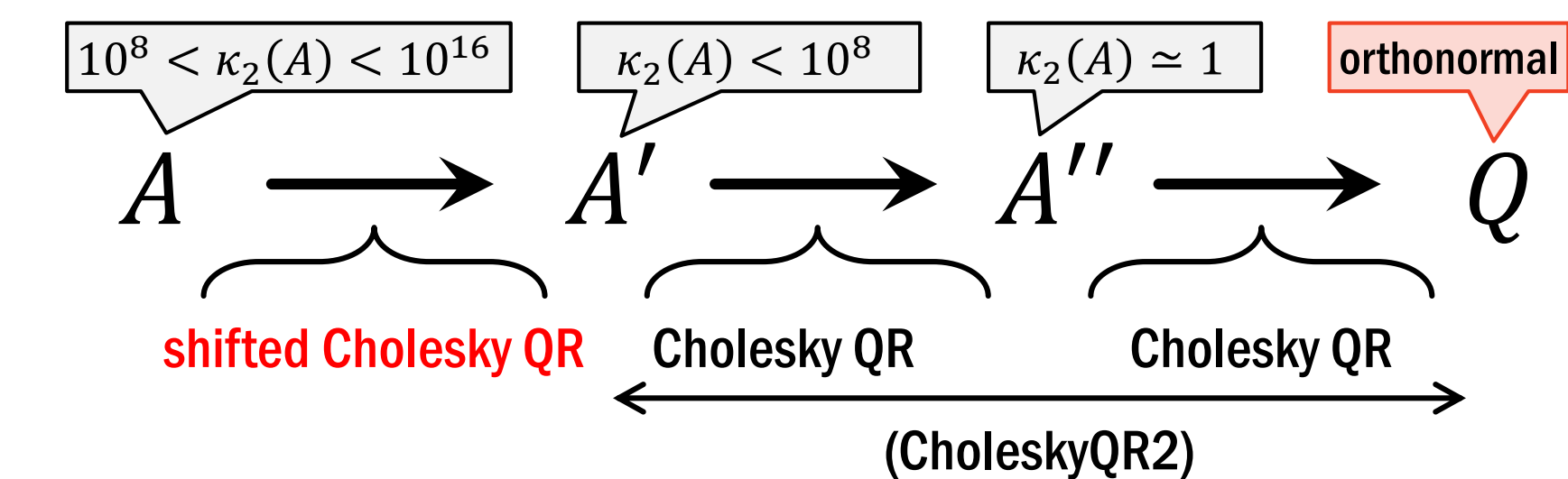
Since Cholesky QR is triangular orthogonalization, if we can find an upper triangular matrix $\tilde{R}$ that satisfies $\kappa_2(A\tilde{R}) < 10^8$, then we can compute the QR factorization of $A\tilde{R}$ using CholeskyQR2 and then obtain $A\tilde{R} = QR \Rightarrow A = Q(R\tilde{R}^{-1})$.

How to find an appropriate upper triangular matrix?

Since if the minimum eigenvalue of a s.p.d. matrix is sufficiently large, its Cholesky factorization succeeds numerically [6], we employ the inverse of the Cholesky factor of $A^T A + sI$ as $\tilde{R}$. We expect that $\kappa_2(A\tilde{R}) \ll \kappa_2(A)$ if s is small.

Proposal of the shifted Cholesky QR algorithm

We proposed the **shifted Cholesky QR** algorithm and employ it as a **preconditioning step** before the CholeskyQR2 algorithm (we call the overall process the **shifted CholeskyQR3** algorithm).



sCholQR(A, s) = [Q, R] // s > 0

- $W := A^T A$
- $R := \text{chol}(W + sI)$
- $Q := AR^{-1}$

The shifted Cholesky QR algorithm

sCholQR3(A, s) = [Q, R] // s > 0

- $[A', R_1] := \text{sCholQR}(A, s)$
- $[Q, R_2] := \text{CholQR2}(A')$
- $R := R_2 R_1$

The shifted CholeskyQR3 algorithm

Theoretical results

Theorem (for details, see [7]):

Assume that roughly $\kappa_2(A) < \mathbf{u}^{-1}$ and let $\mathbf{u}$ is the unit roundoff ($\mathbf{u} \approx 10^{-16}$ in double precision), if we set the shift $s := 11\{mn + n(n+1)\}\mathbf{u}\|A\|_F^2$, then

- $\kappa_2(A') \leq 2\sqrt{1 + \frac{s}{\|A\|_F^2}(\kappa_2(A))^2} \cdot \sqrt{3} \approx O(\sqrt{mn}) \cdot \sqrt{\mathbf{u}} \cdot \kappa_2(A)$,
- $\|Q^T Q - I\|_F \leq O(mn) \cdot \mathbf{u}$ and $\frac{\|QR - A\|_F}{\|A\|_F} \leq O(n^2) \cdot \mathbf{u}$,
where Q, R are the computed results by shifted CholeskyQR3.

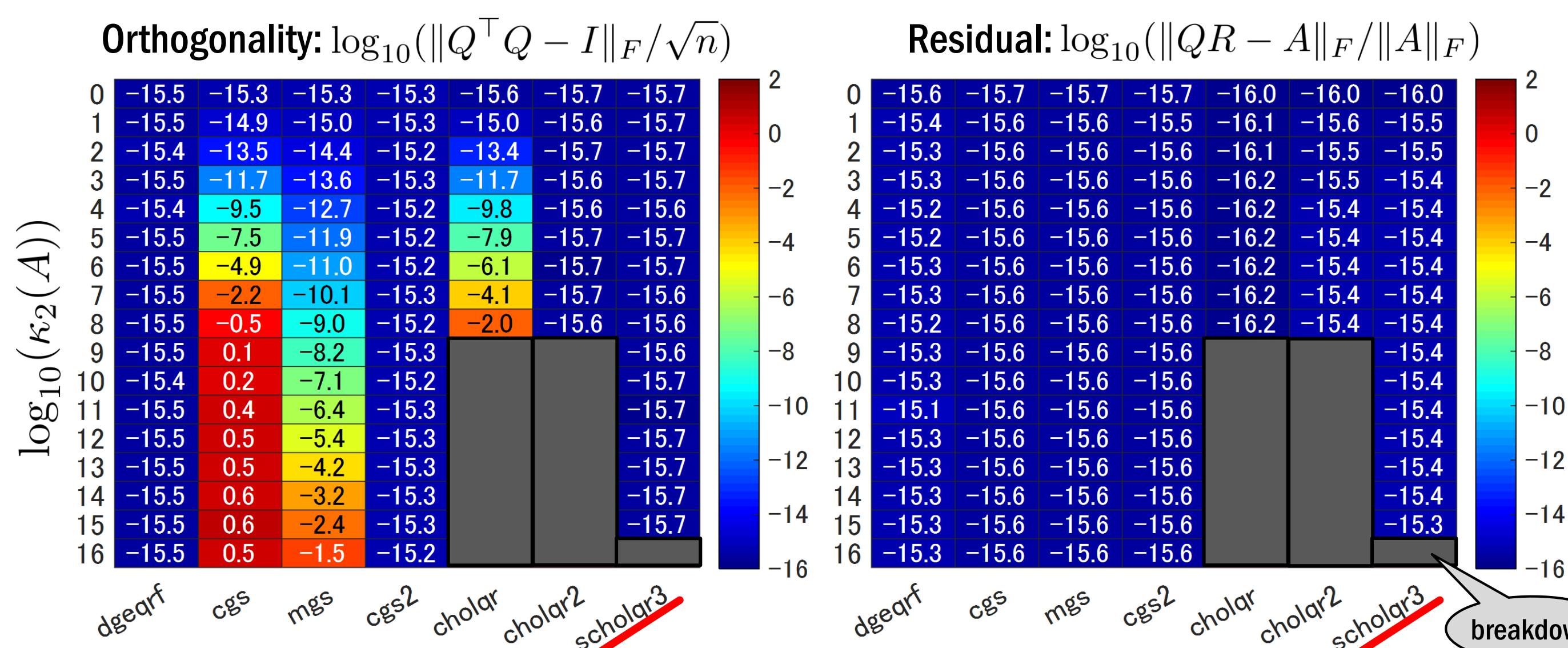
Note: in practical, a smaller shift (e.g. $s := \sqrt{m}\|A\|_F^2 \mathbf{u}$) is usually sufficient.

Performance evaluation

Accuracy

Compute the QR factorization of a matrix generated by $A := U\Sigma V$,

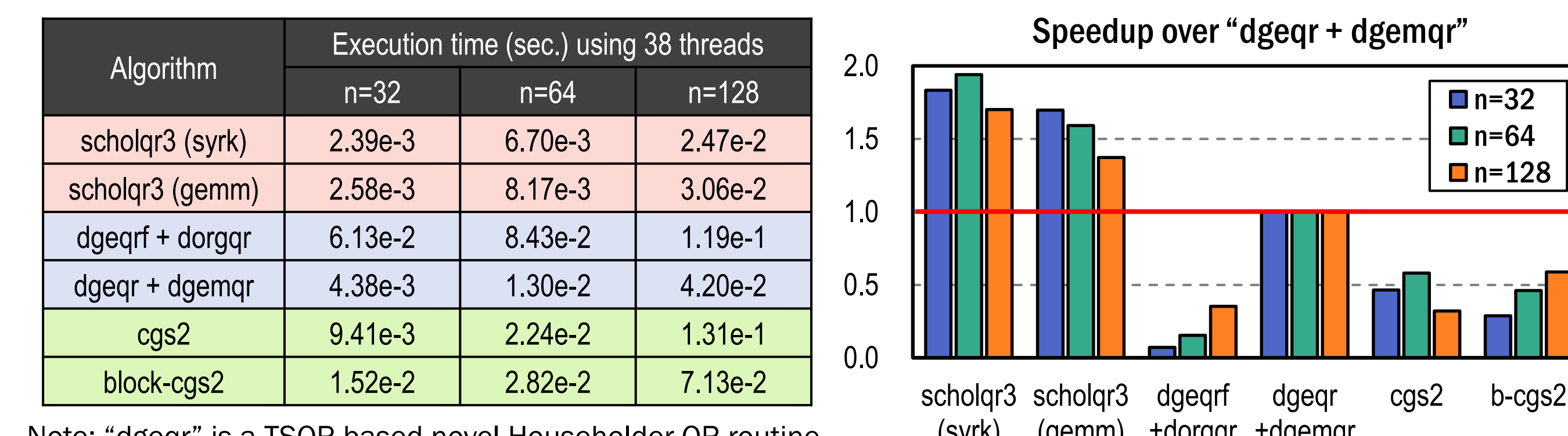
- U, V are random orthogonal matrices, and $\Sigma = \text{diag}(\frac{1}{\sigma}, \frac{1}{\sigma^{n-1}}, \dots, \frac{1}{\sigma^{n-1}}, \sigma)$, $0 < \sigma < 1 \Rightarrow \kappa_2(A) = \frac{1}{\sigma}$,
- $s := \sqrt{m}\|A\|_F^2 \times 10^{-16}$, $m = 100,000$, and $n = 64$.



Note: "dgeqrf" is a traditional Householder QR routine provided in LAPACK ("dorgqr" is used for explicitly forming Q).

Execution time in a recent multicore CPU environment

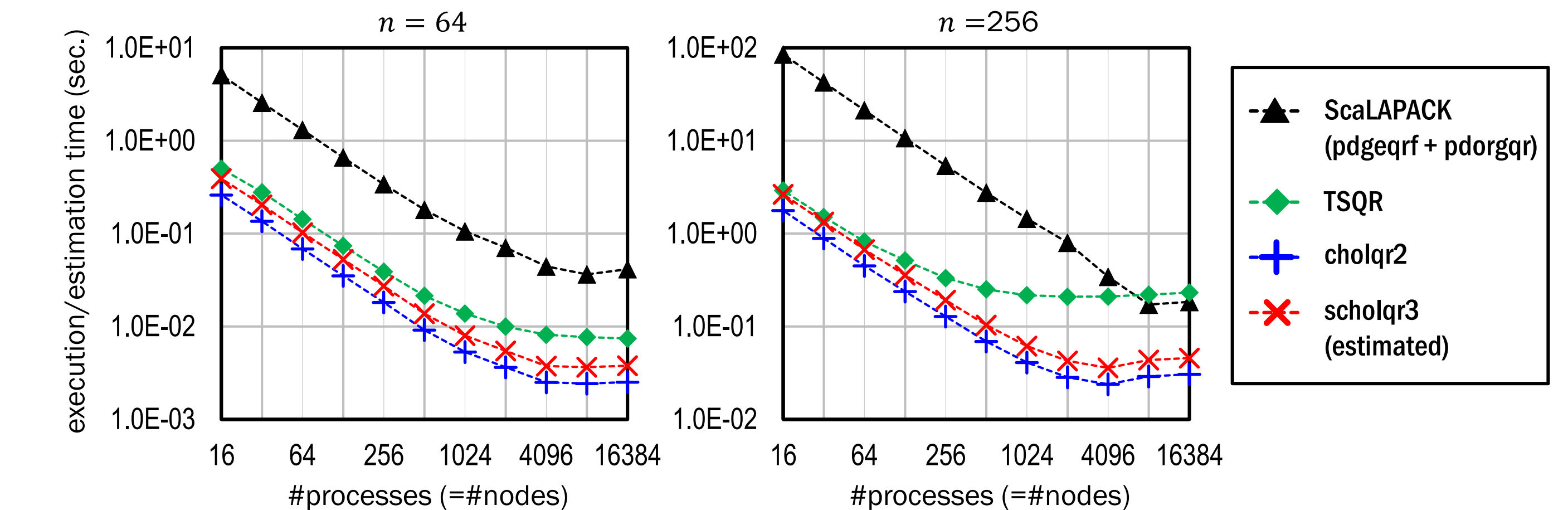
- Environment: one node of Laurel2 system @ ACCMS, Kyoto Univ., Japan (Intel Xeon E5-2695v4 (18-core, 2.1GHz) x2 and 128GB memory).
- Compiler, BLAS, LAPACK: ifort (ver. 17.0.6) and Intel MKL (ver. 2017.0.6 with "-mkl=parallel").
- Settings: $s := \sqrt{m}\|A\|_F^2 \times 10^{-16}$, $m = 100,000$, and $\kappa_2(A) = 10^{14}$.



Note: "dgeqr" is a TSQR-based novel Householder QR routine.

Estimated execution time on a large-scale distributed parallel system

Based on the results of our previous performance evaluation [3] using the K computer, we estimate the execution time of shifted CholeskyQR3 by simply multiplying 1.5 to the measured time of CholeskyQR2. ($m = 4,194,304$)



Extension to oblique inner products

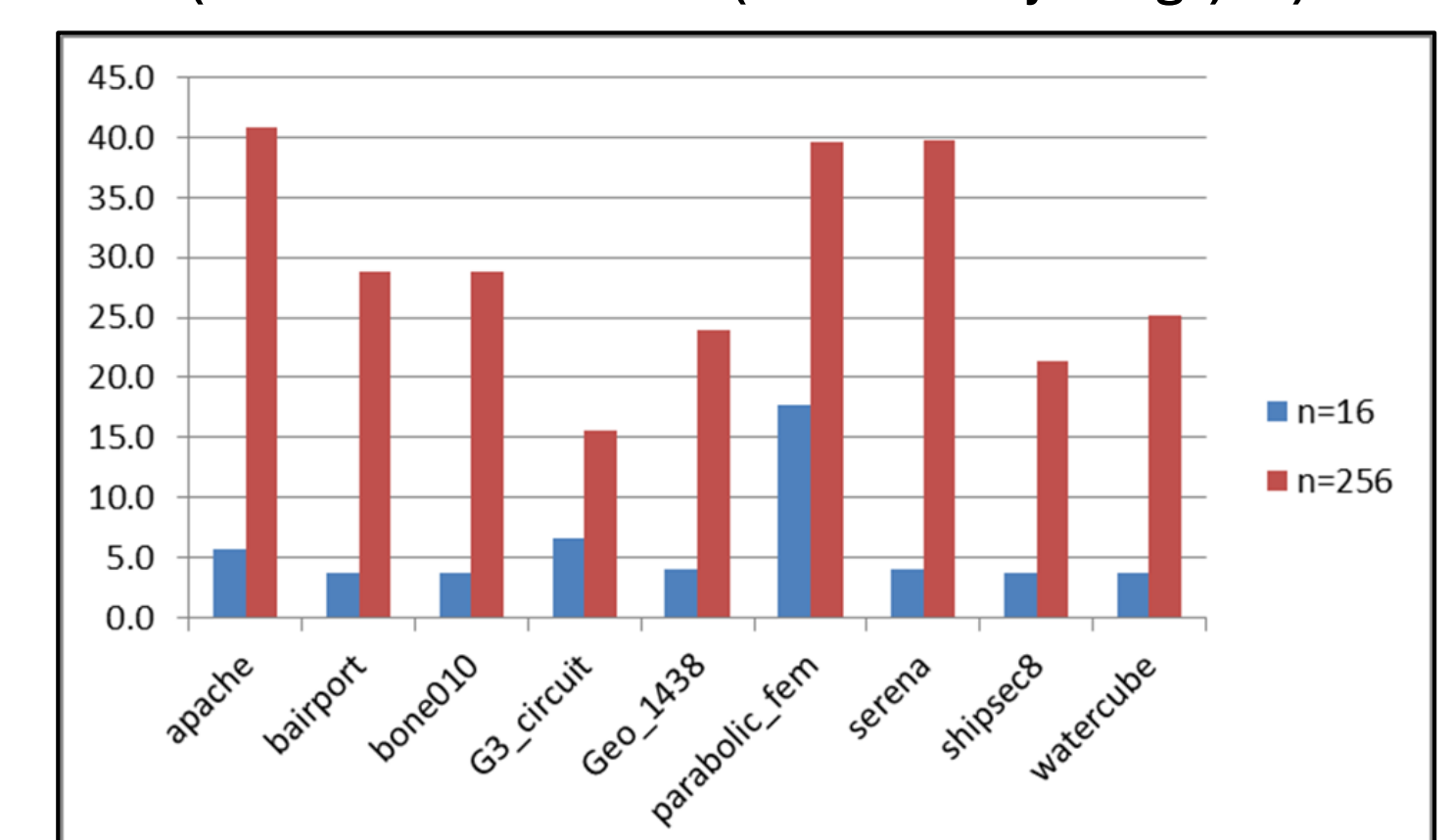
QR factorization in an oblique inner product space

- $Q: B$ -orthogonal ($Q^T B Q = I$), where $B > 0$ (B is symmetric positive definite)
- Inner product is defined as $(x, y)_B := x^T B y$.

Advantages of scholar3

- Since CGS2 is **step-by-step**, sparse matrix vector multiplication (SpMV) is repeatedly required.
- Cholesky QR requires sparse matrix block-vector multiplication ($A^T B A$) only once, which has a **large scope for efficient implementation** [8].
- ShiftedCholeskyQR3 is numerically stable also for QR factorization in an oblique inner product space [7].

Speedup of scholar3 over cgs2 for sparse problems (on Intel Xeon E5-2670 (8-core Sandy Bridge) x2)



(For the details of evaluation, see [9].)

Conclusion

The Shifted CholeskyQR3 algorithm

- uses **shifted Cholesky QR** as a preconditioning step before CholeskyQR2,
- accurately computes the QR factorization of ill-conditioned (roughly $\kappa_2(A) < 10^{16}$) matrices **using only double precision** arithmetic,
- outperforms conventional algorithms in execution time,
- is **simple to implement**,
- is efficient also for QR factorization in an **oblique inner product** space.

References:

- J. Demmel, L. Grigori, M. Hoemmen, J. Langou, Communication-optimal Parallel and Sequential QR and LU Factorizations, SIAM J. Sci. Comp, vol. 34, pp.206-239, 2012.
- A. Stathopoulos, K. Wu, A block orthogonalization procedure with constant synchronization requirements, SIAM J. Sci. Comp, vol. 23, pp. 2165-3282, 2002.
- T. Fukaya, Y. Nakatsukasa, Y. Yanagisawa, Y. Yamamoto, CholeskyQR2: a simple and communication-avoiding algorithm for computing a tall-skinny QR factorization on a large-scale parallel system, Proc. of Scala'14, pp. 31-38, 2014.
- Y. Yamamoto, Y. Nakatsukasa, Y. Yanagisawa, T. Fukaya, Roundoff error analysis of the CholeskyQR2 algorithm, ETNA, vol. 44, pp. 306-326, 2015.
- I. Yamazaki, S. Tomov, J. Dongarra, Mixed-precision Cholesky QR factorization and its case studies on Multicore CPU with Multiple GPUs, SIAM J. Sci. Comp, vol. 37, pp. C307-C330, 2015.
- S. M. Rump, T. Ogita, Super-fast validated solution of linear systems, J. Comput. Appl. Math., vol. 199, pp. 199-206, 2007.
- T. Fukaya, R. Kannan, Y. Nakatsukasa, Y. Yamamoto, and Y. Yanagisawa, Shifted CholeskyQR for computing the QR factorization of ill-conditioned matrices, arxiv:1809.11085, 2018.
- R. Kannan, Efficient sparse matrix multiple-vector multiplication using a bitmapped format, Proc. of HiPC'13, pp. 286-294, 2013.
- R. Kannan, Numerical Linear Algebra problems in Structural Analysis, PhD thesis, The Univ. of Manchester, 2014.