High Performance Computing of Thin QR Decomposition on Parallel Systems

Takeshi Terao *, Katsuhisa Ozaki *, and Takeshi Ogita **

* Shibaura Institute of Technology, ** Tokyo Woman's Christian University

Introduction

- This poster concerns thin QR decomposition for parallel computing.
- Fast algorithms are proposed (ex. TSQR [1], CholeskyQR2 [2] algorithms).
- We introduce CholeskyQR and CholeskyQR2 as follows.

```
% The CholeskyQR algorithm using MATLAB notation.
function [Q, R] = CholQR(A)
    B = A'*A;        % B ≈ A^T A
    R = chol(B);     % Cholesky decomposition of a matrix B
    Q = A/R;         % Solve matrix equation
end
```

```
% The CholeskyQR2 algorithm using MATLAB notation.
function [Q, R] = CholQR2(A)
    [S, R1] = CholQR(A);     % A ≈ S * R1
    [Q, R2] = CholQR(S);     % S ≈ Q * R2
    R = R2*R1;
end
```

- Advantage

✓ Very fast for parallel computing.
✓ These algorithms can benefit from optimized BLAS and LAPACK.

- Disadvantage

✓ If $\kappa_2(A) > \sqrt{u^{-1}}$, Cholesky decomposition for B breaks down in many cases.

➢ u : unit round off. (ex. $u = 2^{-53}$ for binary 64)
➢ $\kappa_2(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$

- Topic

✓ The main topic of our study is to propose robust and fast CholeskyQR algorithms for ill-conditioned matrices.
✓ The proposed algorithms achieve high performance on large-scale parallel systems.

Robust CholeskyQR algorithm

- We focus on thin QR decomposition for full column rank matrices for which CholeskyQR breaks down.
- Cholesky decomposition in CholeskyQR breaks down when a given matrix is ill-conditioned.
- The reason is to take a square root of a negative number in Cholesky decomposition.

```
% The robust CholeskyQR algorithm using MATLAB notation.
function [Q, R] = robust_CholQR(A)
    B = A'*A;                  % B ≈ A^T A
    [R, p] = chol(B);          % If p = 0, B ≈ R^T R, R ∈ R^{n×n}.   If p ≥ 1, B11 ≈ R^T R, R ∈ R^{p×p}.
    if p == 0
        Q = A/R;               % A ≈ QR
    else
        R1 = [R, R'\B12; O, R22];    % where B = [B11, B12; B21, B22]
        S = A/R1;
        [Q, R2] = CholQR(S);
        R = R1*R2;
    end
end
```

R_1 is the preconditioner of CholQR expecting

$$\kappa_2(AR_1^{-1}) \approx \sqrt{u}\kappa_2(A)$$

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$R_1 = \begin{pmatrix} R & R_{12} \\ O & R_{22} \end{pmatrix}$$

- How to set the preconditioner

✓ R_{11} is a computed Cholesky factor of B_{11}.

$$R_{11} = \mathrm{chol}(B_{11}), \quad R_{12} = R_{11}^{-T}B_{12}, \quad R_{22} = \alpha I$$

where

$$\alpha = \min(\sqrt{u}\max(d), \min(d)), d = \mathrm{diag}(R_{11}) \qquad \mathrm{diag}(A) := (a_{11}, a_{22}, \dots, a_{nn})^T$$

```
% The robust CholeskyQR2 algorithm using MATLAB notation.
function [Q, R] = robust_CholQR2(A)
    [S, R1] = robust_CholQR(A);
    [Q, R2] = CholQR(S);
    R = R2*R1;
end
```

- Computational cost

✓ Computational cost for an m-by-n matrix.
✓ P: the number of processors.

	TSQR	CholeskyQR2	Robust CholeskyQR2
#flops	$\frac{4mn^2}{P} + \frac{4}{3}n^3\log_2 P$	$\frac{4mn^2}{P} + n^3$	$\frac{4mn^2}{P} + n^3$ or $\frac{5mn^2 + 3mnp - 2mp^2}{P} + \frac{5n^3}{3}$
#msgs	$2\log_2 P$	$4\log_2 P$	$4\log_2 P$ or $6\log_2 P$
#words	$n^2\log_2 P$	$2n^2\log_2 P$	$2n^2\log_2 P$ or $3n^2\log_2 P$

- Features

✓ The cost of Robust_CholQR2 is nearly the same as that of CholeskyQR2 algorithm, if the CholeskyQR algorithm is successful.
✓ In the worst case, the cost of Robust_CholQR2 is 1.5 times as much as that of CholeskyQR2.

LU-CholeskyQR algorithm [3]

- We have proposed CholeskyQR using LU factors.

```
% The LU-CholeskyQR algorithm using MATLAB notation.
function [Q, R] = LU_CholQR(A)
    [L,U,P] = lu(A);
    B = L'*L;
    S = chol(B);    % Cholesky decomposition for a matrix B
    R = S*U;
    Q = A/R;        % Solve matrix equation
end
```

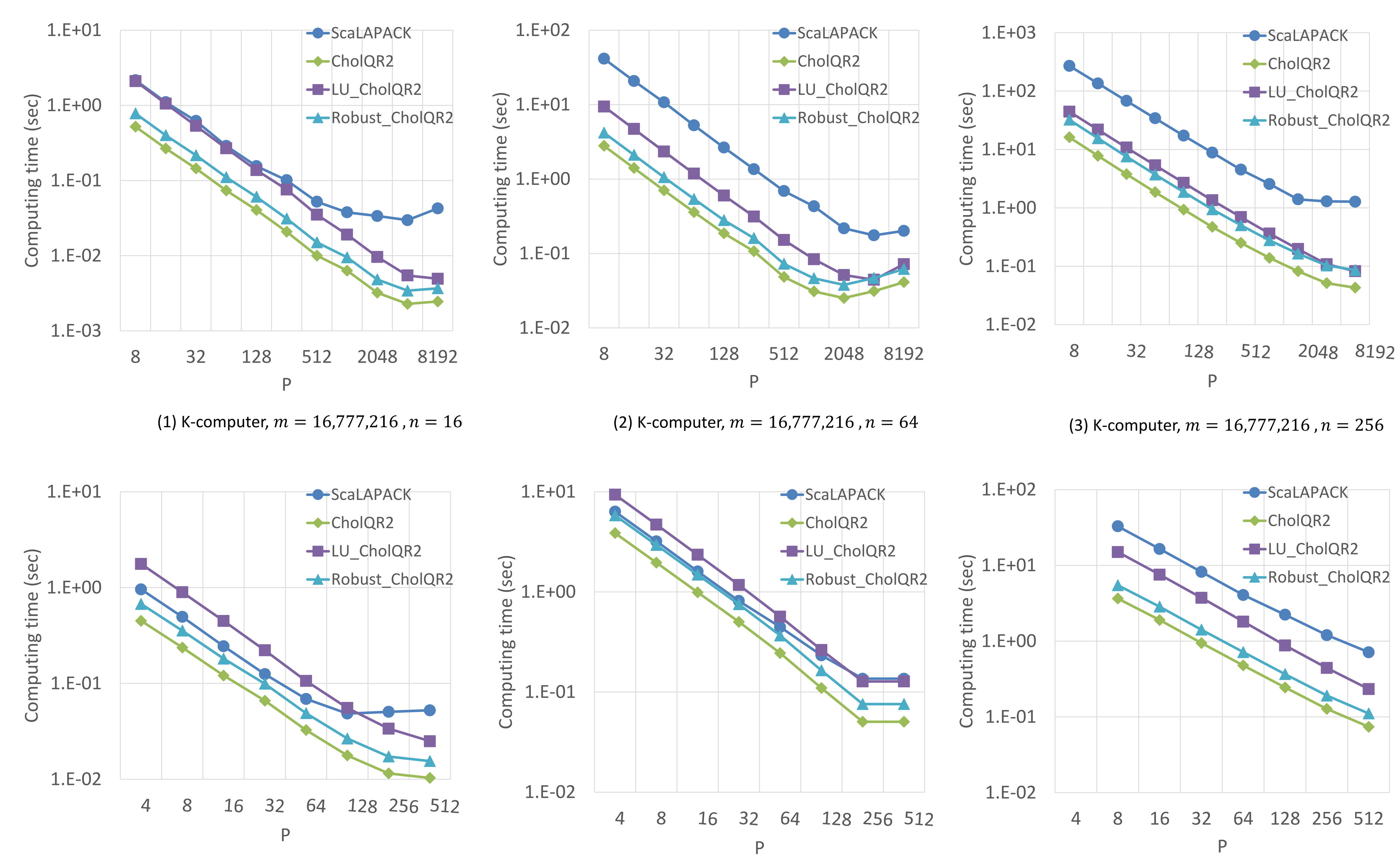
```
% The CholeskyQR2 algorithm using MATLAB notation.
function [Q, R] = LU_CholQR2(A)
    [S, R1] = LU_CholQR(A);
    [Q, R2] = CholQR(S);
    R = R2*R1;
end
```

- Condition number of L is sufficiently small, and Cholesky decomposition for the Gram matrix L^TL runs to completion in many cases.
- Therefore, Cholesky QR through LU decomposition is applicable for ill-conditioned matrices and numerically stable.

Results

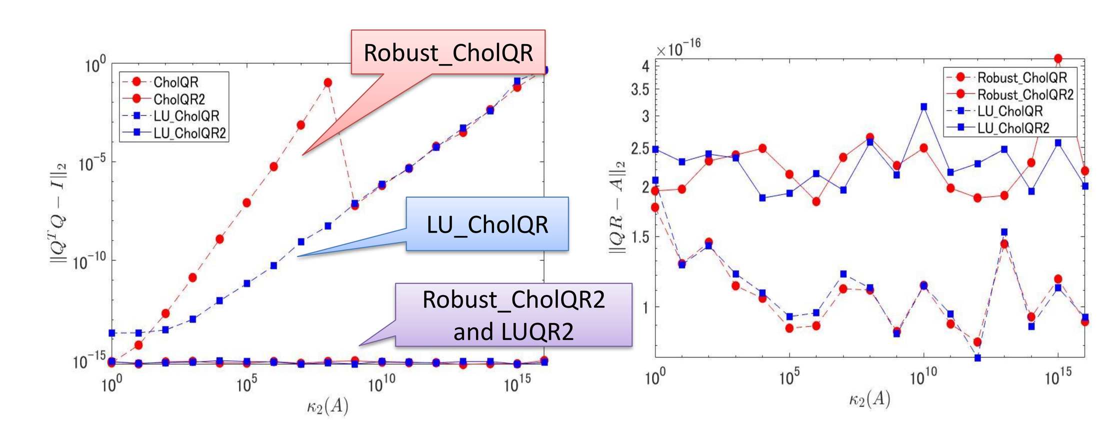
- We compare the computational performance of thin QR decomposition on the RIKEN K-computer and Fujitsu FX100 .

	K-computer	Nagoya FX100
Processor	SPARC64 VIIIfx 8C (2GHz)	Fujitsu SPARC64 Xifx (2.2 GHz)
The number of cores	705,024	92, 024
Memory	1,377,000 GiB	90,000 GiB
Interconnect	Torus fusion (Tofu) interconnect	Tofu interconnect 2
The number of nodes	88,128	2,885



(1) K-computer, m = 16,777,216 , n = 16
(2) K-computer, m = 16,777,216 , n = 64
(3) K-computer, m = 16,777,216 , n = 256
(4) fx100, m = 16,777,216 , n = 16
(5) fx100, m = 16,777,216 , n = 64
(6) fx100, m = 16,777,216 , n = 256

- These figures show the worst case of computing times of Robust_CholQR.
- Robust_CholQR achieves high performance compared to thin-QR decomposition using ScaLAPACK.
- If the number of process of MPI is large or A has large column size, thin LUQR algorithm achieves high performance.



(7) Comparison of orthogonality
(8) Comparison of residual

- $m = 1024, n = 64$.
- Proposed algorithms can compute QR decomposition for ill-conditioned matrices.
- Computed QR factors have good orthogonality and residual
- If $\kappa_2(A) \le \sqrt{u^{-1}}$, we do not compute precondition of Robust CholeskyQR.

Feature of the proposed algorithms

- These algorithms can successfully work for ill-conditioned matrices.
- Computational performances are higher than that of ScaLAPACK (pdgeqrf and pdorgqr).
- For robust CholeskyQR2, the orthogonality is high and the residual is small.

Conclusion

- The proposed CholeskyQR algorithms can run to completion for ill-conditioned matrices.
- Robust CholeskyQR achieves high performance on large-scale parallel systems.

For Robust CholeskyQR

- The cost of robust CholeskyQR2 is the nearly the same as that of CholeskyQR2 algorithm, if the CholeskyQR algorithm works successfully.
- When Cholesky decomposition is breaks down,

✓ If $p \ll n$, cost of the proposed algorithm is 5/4 times as much as that of CholeskyQR2.
✓ If $p \approx n$, cost of the proposed algorithm is 3/2 times as much as that of CholeskyQR2.

- Future works

✓ Reduce computational cost
✓ CholeskyQR algorithm for more ill-conditioned matrices

Acknowledgement

This research was partially supported by MEXT as "Exploratory Issue on Post-K computer"(Development of verified numerical computations and super high performance computing environment for extreme researches).

References

[1] Demmel, J., Grigori, L., Hoemmen, M., & Langou, J. (2014). Communication-avoiding parallel and sequential QR and LU factorizations. In *SIAM Journal of Scientific Computing 34(1), A206-A239* .
[2] Fukaya, T., Nakatsukasa, Y., Yanagisawa, Y., & Yamamoto, Y. (2014, November). CholeskyQR2: a simple and communication-avoiding algorithm for computing a tall-skinny QR factorization on a large-scale parallel system. In *Proceedings of the 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems* (pp. 31-38). IEEE Press.
[3] Terao, T., Ozaki, K. and Ogita, T. LU-Cholesky QR algorithms for thin QR decomposition. Submitted for publication.