

# Mixed Precision Tall and Skinny QR Factorization

**Françoise Tisseur**  
**Department of Mathematics**  
**The University of Manchester**

**Joint work with Srikara Pranesh**

**SIAM CSE21, March 2021**

# Objective

Let  $A \in \mathbb{R}^{m \times n}$ ,  $m \gg n$  have full rank.  $\kappa_2(A) = \|A\|_2 \|A^+\|_2$

**Reduced QR factorization:**  $A = QR$ ,  $Q \in \mathbb{R}^{m \times n}$ ,  $Q^T Q = I_n$ ,  
and  $R \in \mathbb{R}^{n \times n}$  upper triangular.

Want to compute  $\hat{Q}, \hat{R}$  s.t.

$$\|I_n - \hat{Q}^T \hat{Q}\|_2 \leq c_1 u, \quad \|\hat{Q} \hat{R} - A\|_2 \leq c_2 \|A\|_2 u$$

using mixed precision,  $u \leq u_s \leq u_\ell$ .

# Objective

Let  $A \in \mathbb{R}^{m \times n}$ ,  $m \gg n$  have full rank.  $\kappa_2(A) = \|A\|_2 \|A^+\|_2$

**Reduced QR factorization:**  $A = QR$ ,  $Q \in \mathbb{R}^{m \times n}$ ,  $Q^T Q = I_n$ ,  
and  $R \in \mathbb{R}^{n \times n}$  upper triangular.

Want to compute  $\hat{Q}, \hat{R}$  s.t.

$$\|I_n - \hat{Q}^T \hat{Q}\|_2 \leq c_1 u, \quad \|\hat{Q} \hat{R} - A\|_2 \leq c_2 \|A\|_2 u$$

using mixed precision,  $u \leq u_s \leq u_\ell$ .

Two classes of algorithms:

- ▶ orthogonal triangularization (Householder QR, TSQR),
- ▶ triangular orthogonalization (CGS, MGS, **Cholesky-QR**).

# Cholesky-QR Algorithm

$A \in \mathbb{R}^{m \times n}$ ,  $m \gg n$ , full rank.

**CholeskyQR** algorithm:

1  $G = A^T A$

2  $R = \text{chol}(G)$  (upper triang. Cholesky factor of  $G$ )

3  $Q = AR^{-1}$

- ▶ Rich in BLAS 3 operations.
- ▶ Comm. avoiding alg. with similar cost to TSQR.
- ▶ Step 2 may fail when  $\kappa_2(A) \gtrsim u^{-1/2}$ .
- ▶  $\|I_n - Q^T Q\|_2 = O(u\kappa_2(A)^2)$ .

# Cholesky-QR Algorithm

$A \in \mathbb{R}^{m \times n}$ ,  $m \gg n$ , full rank.

**CholeskyQR** algorithm:

1  $G = A^T A$

2  $R = \text{chol}(G)$  (upper triang. Cholesky factor of  $G$ )

3  $Q = AR^{-1}$

- ▶ Rich in BLAS 3 operations.
- ▶ Comm. avoiding alg. with similar cost to TSQR.
- ▶ Step 2 may fail when  $\kappa_2(A) \gtrsim u^{-1/2}$ .
- ▶  $\|I_n - Q^T Q\|_2 = O(u\kappa_2(A)^2)$ .

[Yamazaki, Tomov, Dongarra, SISC 2015] use double-double precision in steps 1–2.

# CholeskyQR2

$A \in \mathbb{R}^{m \times n}$ ,  $m \gg n$ , full rank.

[Yamamoto, Nakatsukasa, Yanagisawa, Fukaya, ETNA 2015]

## CholeskyQR2 algorithm

- 1  $[Q_1, R_1] = \text{CholeskyQR}(A)$
  - 2  $[Q, R_2] = \text{CholeskyQR}(Q_1)$
  - 3  $R = R_2 R_1$ .
- ▶ If  $\kappa_2(A) \lesssim u^{-1/2}$  then  $\kappa_2(Q_1) \approx 1$  and  $\|I_n - Q^T Q\|_2 \approx u$ ,  $\|QR - A\|_2 \approx u\|A\|_2$ .
  - ▶ Communication cost is twice that of TSQR but reduction op. is addition rather than QR fact.
  - ▶ Has the potential to be significantly faster than TSQR.

# ShiftedCholeskyQR3

[Fukaya, Kannan, Nakatsukasa, Yamamoto, Yanagisawa, SISC 2020]

$[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$ :

- 1  $G = A^T A$ .
- 2  $R_0 = \text{chol}(G + sI)$  for some shift  $s > 0$ .
- 3  $Q_0 = AR_0^{-1}$ .

► Shift is for Cholesky to succeed when  $\kappa_2(A) > \sqrt{u^{-1}}$ .

**ShiftedCholeskyQR3** algorithm:

- 1  $[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$
- 2  $[Q_1, R_1] = \text{CholeskyQR}(Q_0)$ ,  $R = R_1 R_0$
- 3  $[Q, R_2] = \text{CholeskyQR}(Q_1)$ ,  $R = R_2 R$

# ShiftedCholeskyQR3

[Fukaya, Kannan, Nakatsukasa, Yamamoto, Yanagisawa, SISC 2020]

$[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$ :

- 1  $G = A^T A$ .
- 2  $R_0 = \text{chol}(G + sI)$  for some shift  $s > 0$ .
- 3  $Q_0 = AR_0^{-1}$ .

► Shift is for Cholesky to succeed when  $\kappa_2(A) > \sqrt{u^{-1}}$ .

**ShiftedCholeskyQR3** algorithm: [Assume  $\kappa_2(A) \approx u^{-1}$ ]

- 1  $[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$  [ $\kappa_2(Q_0) \approx \sqrt{u^{-1}}$ ]
- 2  $[Q_1, R_1] = \text{CholeskyQR}(Q_0)$ ,  $R = R_1 R_0$  [ $\kappa_2(Q_1) \approx O(1)$ ]
- 3  $[Q, R_2] = \text{CholeskyQR}(Q_1)$ ,  $R = R_2 R$



# ShiftedCholeskyQR3

[Fukaya, Kannan, Nakatsukasa, Yamamoto, Yanagisawa, SISC 2020]

$[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$ :

- 1  $G = A^T A$ .
- 2  $R_0 = \text{chol}(G + sI)$  for some shift  $s > 0$ .
- 3  $Q_0 = AR_0^{-1}$ .

► Shift is for Cholesky to succeed when  $\kappa_2(A) > \sqrt{u^{-1}}$ .

**ShiftedCholeskyQR3** algorithm: [Assume  $\kappa_2(A) \approx u^{-1}$ ]

- 1  $[Q_0, R_0] = \text{ShiftedCholeskyQR}(A)$  [ $\kappa_2(Q_0) \approx \sqrt{u^{-1}}$ ]
- 2  $[Q_1, R_1] = \text{CholeskyQR}(Q_0)$ ,  $R = R_1 R_0$  [ $\kappa_2(Q_1) \approx O(1)$ ]
- 3  $[Q, R_2] = \text{CholeskyQR}(Q_1)$ ,  $R = R_2 R$

Steps 1-2 construct **preconditioner**  $R_1 R_0$  s.t.

$Q_1 = A(R_1 R_0)^{-1}$ ,  $\kappa_2(Q_1) \approx 1$  at twice the cost of step 3!

# LU-Cholesky QR

[Terao, Ozaki, Ogita, Parallel Computing 2020] precondition  $A$  with

$[\tilde{Q}, \tilde{R}] = \text{LU-CholeskyQR}(A)$ :

- 1  $[L, U, P] = \text{LU}(A)$  (LU fact. with partial piv.)
- 2  $S = \text{chol}(L^T L)$
- 3  $\tilde{R} = SU, \tilde{Q} = A\tilde{R}^{-1}$ .

[ Note:  $PA = LU$  and  $A^T A = U^T L^T P P^T L U = U^T S^T S U =: \tilde{R}^T \tilde{R} .$  ]

# LU-Cholesky QR

[Terao, Ozaki, Ogita, Parallel Computing 2020] precondition  $A$  with

$[\tilde{Q}, \tilde{R}] = \text{LU-CholeskyQR}(A)$ : [Assume  $\kappa_2(A) \lesssim u^{-1} \approx 10^{16}$ ]

- 1  $[L, U, P] = \text{LU}(A)$  (LU fact. with partial piv.)
- 2  $S = \text{chol}(L^T L)$
- 3  $\tilde{R} = SU, \tilde{Q} = A\tilde{R}^{-1}$ . [ $\kappa_2(\tilde{Q}) \approx 1$ ]

[ Note:  $PA = LU$  and  $A^T A = U^T L^T P P^T L U = U^T S^T S U =: \tilde{R}^T \tilde{R}$  .]

# LU-Cholesky QR

[Terao, Ozaki, Ogita, Parallel Computing 2020] precondition  $A$  with

$[\tilde{Q}, \tilde{R}] = \text{LU-CholeskyQR}(A)$ : [Assume  $\kappa_2(A) \lesssim u^{-1} \approx 10^{16}$ ]

- 1  $[L, U, P] = \text{LU}(A)$  (LU fact. with partial piv.)
- 2  $S = \text{chol}(L^T L)$
- 3  $\tilde{R} = SU, \tilde{Q} = A\tilde{R}^{-1}$ . [ $\kappa_2(\tilde{Q}) \approx 1$ ]

[ Note:  $PA = LU$  and  $A^T A = U^T L^T P P^T L U = U^T S^T S U =: \tilde{R}^T \tilde{R}$  .]

**LU-CholQR2** algorithm:

- 1  $[\tilde{Q}, \tilde{R}] = \text{LU-CholeskyQR}(A)$  (preconditioning)
- 2  $[Q, R_1] = \text{CholeskyQR}(\tilde{Q}), R = R_1 \tilde{R}$

► Similar cost to CholeskyQR3 but not comm. avoiding.

# Preconditioning $A$ in Mixed Precision

Let  $u \leq u_\ell$

(e.g.,  $u$  is rounding error for fp64,  $u_\ell$  is fp32, fp16 or bfloat16).

Assume  $\kappa_2(\text{fl}_\ell(A)) \approx \min\{u_\ell^{-1}, \kappa_2(A)\}$ . The steps

1.  $[L, U, P] = \text{LU}(A)$  in precision  $u_\ell$
2.  $G = L^T L$  in precision  $u_\ell$
3.  $S = \text{chol}(L^T L)$  in precision  $u$
4.  $\tilde{R} = SU$  in precision  $u$

compute preconditioner  $\tilde{R}$  s.t. when applied at precision  $u$ ,

$$\kappa_2(A\tilde{R}^{-1}) \approx \max\{1, u_\ell \kappa_2(A)\}.$$

# Experiment 1, $m = 1000$ , $n = 10$

Preconditioner  $\tilde{R}$  computed in fp16,  $u_\ell = 4.88 \times 10^{-4}$ .

$Q, R$  computed by CholeskyQR applied to  $A\tilde{R}^{-1}$ .

$\text{res} = \|A - QR\|_2 / \|A\|_2$ .

$\kappa_2(A)$	$\kappa_2(\text{fl}_\ell(A))$	$\kappa_2(A\tilde{R}^{-1})$	$\ I - Q^T Q\ _2$	res
1.0e+2	1.0e+2	1.3e+0	2.7e-16	1.8e-16
1.0e+3	1.0e+3	1.3e+0	6.7e-16	1.4e-16
1.0e+4	8.9e+3	1.3e+0	4.7e-16	2.2e-16
1.0e+5	3.6e+4	3.4e+0	1.1e-15	1.2e-16
1.0e+6	2.4e+4	2.6e+1	1.8e-14	1.8e-16
1.0e+7	3.0e+4	4.3e+2	9.0e-12	9.9e-17
1.0e+8	3.3e+4	2.4e+3	5.6e-11	1.1e-16

$$\kappa_2(A\tilde{R}^{-1}) \approx \max\{1, u_\ell \kappa_2(A)\}.$$

# Precondition $A$ in Three Precisions

Let  $u \leq u_s \leq u_h$  (double/single/half).

- 1  $Q = A, R = I$
- 2 for iter = 1:4
- 3      $R_h = \text{LU-Chol}(Q, u_h)$  in precision  $u_h$
- 4     Update  $R \leftarrow R_h R$  in precision  $u$
- 5     if **estimate of  $\kappa_2(Q) < cu_h^{-1}$ , break**,end
- 6      $Q = AR^{-1}$  in precision  $u_s$  if iter=1 and  $u$  otherwise
- 7 end

- ▶ Return  $R$  such that  $\kappa_2(AR^{-1}) \approx 1$ .
- ▶ No knowledge of  $\kappa_2(A)$  is required.
- ▶ Estimate of  $\kappa_2(Q)$  relies on  $\hat{R}$ .

# Experiment 2, $m = 1000$ , $n = 10$

$$u = 1.11 \times 10^{-16}, u_s = 5.96 \times 10^{-8}, u_h = 4.88 \times 10^{-4}.$$

$$\text{res} = \|A - QR\|_2 / \|A\|_2.$$

$\kappa_2(A)$	iter	$\kappa_2(A\tilde{R}^{-1})$	$\ I - Q^T Q\ _2$	res
1.0e+2	1	1.2	4.9e-16	1.7e-16
1.0e+3	1	1.4	8.9e-16	1.6e-16
1.0e+4	2	1.3	4.5e-16	1.7e-16
1.0e+5	2	1.2	2.6e-16	1.9e-16
1.0e+6	2	1.6	4.5e-16	1.6e-16
1.0e+7	2	1.4	5.9e-16	1.4e-16
1.0e+8	2	2.8	9.0e-16	1.2e-16
1.0e+9	3	1.3	7.8e-16	1.3e-16
1.0e+10	3	1.5	4.7e-16	1.4e-16
1.0e+12	4	1.5	4.5e-16	1.3e-16
1.0e+13	4	1.3	6.7e-16	1.3e-16



# Summary

- ▶ Cholesky-QR requires a preconditioner  $\tilde{R}$  s.t.  $\kappa(A\tilde{R}^{-1}) \approx 1$ .
- ▶ ShiftedCholeskyQR+CholeskyQR precondition and LU-CholeskyQR precondition are twice the cost of Cholesky-QR.
- ▶ Propose to use mixed-precision to compute an LU-CholeskyQR preconditioner (expected to be 4 times faster when  $\kappa(A) \lesssim 10^4$ ).
- ▶ For practical implementations of preconditioned Cholesky-QR, need a good estimate of  $\kappa(A)$ .