Introduction

Target problem: QR factorization of a tall-skinny matrix

- A: m x n real and full column rank (mainly focus on m >> n)
- Q: m x n orthogonal
- R: n x n upper triangular

\[ A = QR \]

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).

\[ s \text{CholQR}(A,s) = [Q,R]//s > 0 \]

1. \( s = \sqrt[n]{\lambda_{max}} \) A
2. \( R = \text{chol}(W + sI) \)
3. \( Q = AR^{-1} \)

Orthogonal triangularization \((Q_0 \cdots Q_n; A =: A)\)

Triangular orthogonalization \((A_0 R_1 \cdots R_n =: Q)\)

The shifted CholeskyQR3 algorithm

\[ \text{CholQR2}^3 = \text{sCholQR} + \text{QR factorization in an oblique inner product space} \]

Theoretical results

Theorem (for details, see [7]):

Assume that roughly \( \kappa_2(A) < w^{-2} \) and let \( w \) be the unit roundoff (\( w = 10^{-16} \) in double precision).

If we set the shift \( s = 11(n + m + 1)\|A\|_F, \) then

\[ \kappa_2(Q) \leq 2 \left( 1 + \frac{1}{m} \right) \kappa_2(A) \]

\[ \|Q\|_2 \approx \|R\|_2 \approx \|A\|_2 \]

Note: in practical, a smaller shift (e.g. \( s = \sqrt[10]{\|A\|_F} \)) is usually sufficient.

Accuracy

Compute the QR factorization of a matrix generated by \( A = UZV' \),

- \( U, V \) are random orthogonal matrices, and \( Z = \text{diag}(1, 1, 1, \cdots, 1, e^{-\alpha} \cdots, e^{-\alpha}) \), \( 0 < \alpha < 1 \)

- \( x = \text{vec}(A) \times 10^{18}, m = 100,000, \) and \( n = 64 \).

Performance evaluation

Execution time in a recent multicore CPU environment

- Environment: one node of Laure8 system @ ACCMS, Kyoto Univ., Japan (Intel Xeon ES-2695V4 x16, 128GB RAM x2 and 129GB64 memory)

- Compiler, BLAS, LAPACK: ifort (ver. 17.0.6) and Intel MKL (ver. 2017.0.6 with -mkl-parallel)

- Settings: \( s = \sqrt[10]{\|A\|_F} \times 10^{18}, m = 100,000, \) and \( \kappa_2(A) = 10^{44} \).

Execution time in 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, \( n = 4, 194,304 \).

Extension to oblique inner products

QR factorization in an oblique inner product space

- Q: orthogonal (\( Q^TQ = I \)), where \( B > 0 \) (is symmetric positive definite)

- Inner product is defined as \((x,y) = x^TBy\).

Advantages of scholqr3

- Since CGS2 is step-by-step, sparse matrix vector multiplication (SpMV) is repeatedly required.

- CholeskyQR requires sparse matrix block-vector multiplication (\( A'BA \)) 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].

Conclusion

The Shifted CholeskyQR3 algorithm

- uses shifted CholeskyQR 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:


Performance evaluation of the shifted Cholesky QR algorithm for ill-conditioned matrices

Takeshi Fukaya (Hokkaido Univ., fukaya@iic.hokudai.ac.jp), Ramaseshan Kannan (Arup UK), Yuji Nakatsukasa (NII), Yusaku Yamamoto (UEC), Yuka Yanagisawa (Waseda Univ.)