Solving Total Least Squares Problems Using Mixed Precision

SIAM CSE23

Eda Oktay, Erin Carson

Department of Numerical Mathematics, Charles University

March 2, 2023

. This study is funded from the Charles University Research Program No. UNCE/SCI/023, GAUK project No. 202722, and the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Least-Squares Problems (LSP)

Goal : Solve $min_x \|b - Ax\|_2$, where $A \in \mathbb{R}^{m \times n}$ $(m \geq n)$ has rank n

Solve : QR factorization - $A = QR = [Q_1 \quad Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix}$, where $Q \in \mathbb{R}^{m \times m}$ is orthogonal, and $U \in \mathbb{R}^{n \times n}$ is upper triangular.

$$Ax = b \rightarrow [Q_1 \quad Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix} x = b \rightarrow x = U^{-1} Q_1^T b, \|b - Ax\|_2 = \|Q_2^T b\|_2$$

Normal Equations : $A^T Ax = A^T b \rightarrow \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$

Mixed Precision LSP

Mixed precision IR is often used to solve least squares problems.

- Solve $Ax = b$
- Improve accuracy of the computed approximate solution $\hat{x}$
- Compute x_0 by GEPP

Algorithm: IR [Wilkinson, 1963]

$Ax_0 = b$ in precision u_f ; store in precision u

for $i = 0 : i_{max} - 1$ **do**

  $r_i = b - Ax_i$ in precision u_r ; store in precision u

  $Ad_{i+1} = r_i$ in precision u_s ; store in precision u

  $x_{i+1} = x_i + d_{i+1}$ in precision u

  if *converged* **then**

    return x_{i+1}

  end

end

Mixed Precision LSP

Mixed precision IR solves augmented system or normal equations.

Augmented system :

- [Björck, 1967] : Two precision IR
- [Demmel et al., 2009] : Two precision LSIR with BLAS
- [Carson et al., 2020] : Three precision GMRES-LSIR

Normal equations :

- [Yamazaki et al., 2015] : Mixed precision Cholesky
- [Higham and Pranesh, 2021] : Low precision Cholesky preconditioner in GMRES-LSIR and CG-LSIR

Total Least Squares Problems (TLSP)

Standard linear model : $Ax = b + r$, r- random error vector

Total Least Squares Problems (TLSP)

Standard linear model : $Ax = b + r$, r- random error vector

How realistic is this ?

Total Least Squares Problems (TLSP)

Standard linear model : $Ax = b + r$, r- random error vector

How realistic is this ?

Errors-in variables model : $(A + E)x = b + r$, E- random error matrix

Total Least Squares Problems (TLSP)

The first numerically stable algorithm ([Golub and Van Loan, 1980]).

$$min_{E,r} \|[E; r]\|_F, \ (A + E)x = b + r$$

How to Solve TLSP ?

- **Direct methods :** Singular value decomposition

  $[A; b] = U\Sigma V^T$, $\Sigma =$diag$(\sigma_1, \cdots, \sigma_{n+1})$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+1} \geq 0$ and $V = [v_1, \cdots, v_{n+1}]$.

  Assume rank(A)=n. If $\sigma_{n+1} = 0$, then $[E; r] = 0$. If $\sigma_{n+1} > 0$, then $min_{\text{rank}(A+E, b+r)<n+1} \|[E; r]\|_F = \sigma_{n+1}$ and

  $$x_{TLS} = -\frac{1}{v_{n+1,n+1}}[v_{1,n+1}, \cdots, v_{n,n+1}]^T.$$

How to Solve TLSP ?

- **Direct methods :** Singular value decomposition

  $[A; b] = U\Sigma V^T$, $\Sigma =$diag$(\sigma_1, \cdots, \sigma_{n+1})$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n+1} \geq 0$ and $V = [v_1, \cdots, v_{n+1}]$.

  Assume rank(A)=n. If $\sigma_{n+1} = 0$, then $[E; r] = 0$. If $\sigma_{n+1} > 0$, then $min_{\text{rank}(A+E, b+r)<n+1} \|[E; r]\|_F = \sigma_{n+1}$ and $x_{TLS} = -\frac{1}{v_{n+1,n+1}}[v_{1,n+1}, \cdots, v_{n,n+1}]^T$.

- **Iterative methods :** Inverse iteration, (inverse) Chebyshev iteration [Van Huffel, 1991], Rayleigh-quotient iteration (RQI) [Björck, 1997], block Golub-Kahan bidiagonalization [Hnětynková et al., 2013]

Rayleigh-Quotient Iteration (RQI)

Algorithm: RQI

Given $x^{(0)}$ with $\|x^{(0)}\| = 1$
$\rho^{(0)} = (x^{(0)})^T B x^{(0)}$
for k=1,2,... **do**
  $(B - \rho^{(k-1)} I)\omega = x^{(k-1)}$
  $x^{(k)} = \omega / \|\omega\|$
  $\rho^{(k)} = (x^{(k)})^T B x^{(k)}$
end

- Inverse iteration with shift ρ - Rayleigh quotient to estimate eigenpair
- Given eigenvector x of B, estimate the corresponding eigenvalue.
- Rayleigh quotient - $\rho(x) = \dfrac{x^T B x}{x^T x}$
- Continually improve eigenvalue estimates to increase the rate of convergence of inverse iteration at each step.

RQI for TLSP

Eigenvalue problem : $\lambda = \sigma_{n+1}^2$ and $x = x_{TLS}$ satisfy

$$\begin{bmatrix} A^T A & A^T b \\ b^T A & b^T b \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} = \lambda \begin{bmatrix} x \\ -1 \end{bmatrix}$$

or

$$\begin{bmatrix} A^T \\ b^T \end{bmatrix} (-r) = \lambda \begin{bmatrix} x \\ -1 \end{bmatrix}, \text{ with } r = b - Ax.$$

Newton's method : $\begin{bmatrix} f(x,\lambda) \\ g(x,\lambda) \end{bmatrix} = \begin{bmatrix} -A^T r - \lambda x \\ -b^T r + \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Normal equations : $(A^T A - \sigma_{n+1}^2 I)x = A^T b$

RQI for TLSP

Rayleigh quotient : $\rho(x) = \dfrac{(x^T A^T - b^T)(Ax - b)}{x^T x + 1} = \dfrac{r^T r}{x^T x + 1}$

The next approximation : $\begin{bmatrix} J^{(k)} & A^T b \\ b^T A & \eta_k \end{bmatrix} \begin{bmatrix} x^{(k+1)} \\ -1 \end{bmatrix} = \beta_k \begin{bmatrix} x^{(k)} \\ -1 \end{bmatrix},$

with $J^{(k)} = A^T A - \rho_k I$, $\eta_k = b^T b - \rho_k$, β_k - scaling factor

RQI for TLSP

Algorithm: RQI for TLSP [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Solve $A^T Au = x$
$x = x + \rho^2 u$
for *k=1,2,...* **do**
    $r = b - Ax$
    $\rho^2 = r^T r/(1 + x^T x)$
    $f = -A^T r - \rho^2 x$
    $g = -b^T r + \rho^2$
    solve $(A^T A - \rho^2 I)\omega = -f$
    $z = x + \omega$
    $\beta = (z^T f - g)/(z^T x + 1)$
    solve $(A^T A - \rho^2 I)u = x$
    $x = z + \beta u$
end

RQI for TLSP

Algorithm: RQI for TLSP [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Solve $A^T A u = x$
$x = x + \rho^2 u$

← One step of inverse iteration

for *k=1,2,...* **do**
 $r = b - Ax$
 $\rho^2 = r^T r/(1 + x^T x)$
 $f = -A^T r - \rho^2 x$
 $g = -b^T r + \rho^2$
 solve $(A^T A - \rho^2 I)\omega = -f$
 $z = x + \omega$
 $\beta = (z^T f - g)/(z^T x + 1)$
 solve $(A^T A - \rho^2 I)u = x$
 $x = z + \beta u$
end

RQI for TLSP

Algorithm: RQI for TLSP [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Solve $A^T A u = x$
$x = x + \rho^2 u$

← One step of inverse iteration

for k=1,2,... **do**
  $r = b - Ax$
  $\rho^2 = r^T r/(1 + x^T x)$
  $f = -A^T r - \rho^2 x$
  $g = -b^T r + \rho^2$
  solve $(A^T A - \rho^2 I)\omega = -f$
  $z = x + \omega$
  $\beta = (z^T f - g)/(z^T x + 1)$
  solve $(A^T A - \rho^2 I)u = x$
  $x = z + \beta u$
end

How to solve ?

RQI-PCGTLS

If $(A^T A - \rho^2 I)$ is SPD $\rightarrow$ Use conjugate gradient ! (with $k + 1$ iterations)

Preconditioner : Cholesky factor R of $A^T A$

RQI-PCGTLS

If $(A^T A - \rho^2 I)$ is SPD $\rightarrow$ Use conjugate gradient ! (with $k+1$ iterations)

Preconditioner : Cholesky factor R of $A^T A$

Rayleigh Quotient Iteration with Preconditioned Conjugate Gradient for TLS problems (RQI-PCGTLS) [Björck et al., 2000]

RQI-PCGTLS

Algorithm: PCGTLS for $(A^TA - \rho^2 I)\omega = f$ [Björck et al., 2000]

Initialize $\omega^{(0)} = 0$, $p^{(0)} = s^{(0)} = R^{-T}f$, $\eta_0 = \|s^{(0)}\|_2^2$
for *j=0,1,...,l, while* $\delta_j \neq 0$ **do**

$q^{(j)} = R^{-1}p^{(j)}$
$\delta_j = \|p^{(j)}\|_2^2 - \rho^2\|q^{(j)}\|_2^2$
$\alpha_j = \eta_j/\delta_j$
$\omega^{(j+1)} = \omega^{(j)} + \alpha_j q^{(j)}$
$s^{(j+1)} = s^{(j)} - \alpha_j(p^{(j)} - \rho^2 q^{(j)})$
$\eta_{j+1} = \|s^{(j+1)}\|_2^2$
$\beta_j = \eta_{j+1}/\eta_j$
$p^{(j+1)} = s^{(j+1)} + \beta_j p^{(j)}$

end

A is not used at all!

Assuming *R* is computed exactly, $\tilde{C} = I - \rho^2 R^{-T}R^{-1}$ is the preconditioned matrix.

Rayleigh-Quotient Iteration (RQI)

Algorithm: RQI-PCGTLS [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Compute $A^T A$ and its Cholesky factor
Solve $A^T Au = x$
$x = x + \rho^2 u$
for *k=1,2,...* **do**
  $r = b - Ax$
  $\rho^2 = r^T r/(1 + x^T x)$
  $f = -A^T r - \rho^2 x$
  $g = -b^T r + \rho^2$
  solve $(A^T A - \rho^2 I)\omega = -f$
  $z = x + \omega$
  $\beta = (z^T f - g)/(z^T x + 1)$
  solve $(A^T A - \rho^2 I)u = x$
  $x = z + \beta u$
end

$A^T A$ and Cholesky can be expensive !

Rayleigh-Quotient Iteration (RQI)

Algorithm: RQI-PCGTLS [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Compute $A^T A$ and its Cholesky factor
Solve $A^T Au = x$
$x = x + \rho^2 u$
for k=1,2,... **do**
    $r = b - Ax$
    $\rho^2 = r^T r/(1 + x^T x)$
    $f = -A^T r - \rho^2 x$
    $g = -b^T r + \rho^2$
    solve $(A^T A - \rho^2 I)\omega = -f$
    $z = x + \omega$
    $\beta = (z^T f - g)/(z^T x + 1)$
    solve $(A^T A - \rho^2 I)u = x$
    $x = z + \beta u$
end

$A^T A$ and Cholesky can be expensive !

Use lower precision !

Algorithm: RQI-PCGTLS [Björck et al., 2000]

$x = x_{LS}$
$r = b - Ax$
$\rho^2 = r^T r/(1 + x^T x)$
Compute $A^T A$ and its Cholesky factor
Solve $A^T A u = x$
$x = x + \rho^2 u$
for *k=1,2,...* **do**
    $r = b - Ax$
    $\rho^2 = r^T r/(1 + x^T x)$
    $f = -A^T r - \rho^2 x$
    $g = -b^T r + \rho^2$
    Solve $(A^T A - \rho^2 I)\omega = -f$ with PCGTLS
    $z = x + \omega$
    $\beta = (z^T f - g)/(z^T x + 1)$
    Solve $(A^T A - \rho^2 I)u = x$ with PCGTLS
    $x = z + \beta u$
end

Algorithm: RQI-PCGTLS-MP

$x = x_{LS}$

$r = b - Ax$ in precision u_r

$\rho^2 = r^T r/(1 + x^T x)$

Compute $A^T A$ and its Cholesky factor in precision u_c

Solve $A^T A u = x$

$x = x + \rho^2 u$

for *k=1,2,...* **do**

  $r = b - Ax$

  $\rho^2 = r^T r/(1 + x^T x)$ in precision u

  $f = -A^T r - \rho^2 x$

  $g = -b^T r + \rho^2$

  Solve $(A^T A - \rho^2 I)\omega = -f$ with PCGTLS in precision u_p

  $z = x + \omega$

  $\beta = (z^T f - g)/(z^T x + 1)$

  Solve $(A^T A - \rho^2 I)u = x$ with PCGTLS in precision u_p

  $x = z + \beta u$

end

Mixed Precision RQI-PCGTLS (RQI-PCGTLS-MP)

Use four precision in RQI-PCGTLS :

u_r - residual precision,

u - RQI working precision (store the data and solution),

u_p - PCGTLS working precision (store the data and solution),

u_c - $A^T A$ and Cholesky precision,

with $u_r \leq u \leq u_p \leq u_c$.

In this study we are using (u_r, u, u_p, u_c) = (quad,double,single,half).

Mixed Precision RQI-PCGTLS (RQI-PCGTLS-MP)

Termination criteria : Stop when the normalized residual norm increases ($\gamma_{k+1} > \gamma_k$) :

$$\gamma_k = \left(\frac{\|f_k\|_2^2 + g_k^2}{\|x^{(k)}\|_2^2 + 1} \right)^{1/2}$$

Necessary conditions for RQI-PCGTLS : Assume $\sigma_1' \geq \sigma_2' \geq \ldots \geq \sigma_n' \geq 0$ are the singular values of A. Then,

[Björck, 1996] : $\sigma_n' > \sigma_{n+1} \Rightarrow x_{TLS}$ is unique.

[Björck et al., 2000] : $A^T A - \rho^2 I$ is SPD $\Rightarrow$ PCGTLS is convergent.

Mixed Precision RQI-PCGTLS (RQI-PCGTLS-MP)

Termination criteria : Stop when the normalized residual norm increases ($\gamma_{k+1} > \gamma_k$) :

$$\gamma_k = \left(\frac{\|f_k\|_2^2 + g_k^2}{\|x^{(k)}\|_2^2 + 1} \right)^{1/2}$$

Necessary conditions for RQI-PCGTLS-MP : Assume $\sigma_1' \geq \sigma_2' \geq \ldots \geq \sigma_n' \geq 0$ are the singular values of A. Then,

[Björck, 1996] : $\sigma_n' > \sigma_{n+1} \Rightarrow x_{TLS}$ is unique.

[Björck et al., 2000] : $A^T A - \rho^2 I$ is SPD $\Rightarrow$ PCGTLS is convergent.

Using [Demmel, 1989, Higham and Pranesh, 2021] :

$$\frac{\sigma_n'^2}{(2\sigma_n'^2 + n)(n+1)} > u_c, \quad \text{where } A^T A \in \mathbb{R}^{n \times n}.$$

Numerical Experiments - Van Huffel matrix, $n = 100$

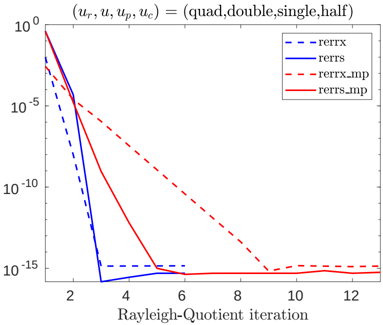
[Van Huffel and Vandewalle, 1991]

$(A + E)x = b + e, \quad \epsilon = 10^{-6}$

E - $\epsilon \times$ random matrix

e - $\epsilon \times$ random vector

$$\underbrace{\begin{bmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ -1 & -1 & \cdots & n-1 \\ -1 & -1 & \cdots & -1 \\ -1 & -1 & \cdots & -1 \end{bmatrix}}_{n \times n-2} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-3} \\ x_{n-2} \end{bmatrix} \approx \begin{bmatrix} -1 \\ -1 \\ \cdot \\ \cdot \\ \cdot \\ -1 \\ n-1 \\ -1 \end{bmatrix}$$



Numerical Experiments - Toeplitz matrix, $n = 100$

$(\bar{T} + E)x = \bar{g} + e, \quad \epsilon = 10^{-6}$

$\gamma_{\bar{T}} = 0.001\|\bar{T}\|, \quad \gamma_{\bar{g}} = 0.001\|\bar{g}\|$

$\bar{g}$ - vector of ones,

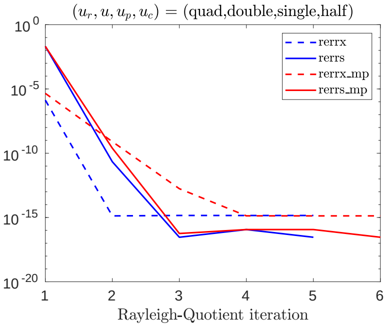
E - $\epsilon \times \gamma_{\bar{T}} \times$ random Toeplitz matrix,

e - $\epsilon \times \gamma_{\bar{g}} \times$ random vector

$\bar{T}^{n\times(n-2\omega)}$ - lower Toeplitz matrix

$$t_{i,1} = \begin{cases} \frac{1}{\sqrt{2\pi\alpha^2}} e^{\left[\frac{-(\omega-i+1)^2}{2\alpha^2}\right]} & i = 1, 2, \ldots, 2\omega + 1, \\ 0 & \text{otherwise}, \end{cases}$$

$$t_{1,j} = \begin{cases} t_{1,1} & \text{if } j = 1, \\ 0 & \text{otherwise}, \end{cases} \quad \text{with} \quad \alpha = 1.25,\ \omega = 2$$



Conclusion

- Forming $A^T A$ and Cholesky factorization can dominate the computational cost.
- Use as low precision as possible to decrease the computational cost.
- The precision for factorization depends on σ'_n.
- RQI-PCGTLS-MP converges to the same level as RQI-PCGTLS
- Using low precision increases the number of RQI

Conclusion

- Forming $A^T A$ and Cholesky factorization can dominate the computational cost.
- Use as low precision as possible to decrease the computational cost.
- The precision for factorization depends on σ'_n.
- RQI-PCGTLS-MP converges to the same level as RQI-PCGTLS
- Using low precision increases the number of RQI but performance can compensate the additional iterations !

Next Step...

- Performance analysis with larger sized matrices
- Complete theoretical analysis of RQI-PCGTLS-MP by adapting, e.g.,
    - Results on inexact Newton in [Higham and Mary, 2022, Tisseur, 2001, Kelley, 2022],
    - Results on inexact RQI in [Freitag and Spence, 2007, Simoncini and Eldén, 2002].

Complete analysis includes :

- Proof of attainable accuracy,
- Theoretical explanation of different convergence delay.

Thank you for your attention !

oktay@karlin.mff.cuni.cz

Björck, Å. (1967).
Iterative refinement of linear least squares solutions i.
BIT Numerical Mathematics, 7(4) :257–278.

Björck, Å. (1996).
Numerical methods for least squares problems.
SIAM.

Björck, Å. (1997).
Newton and rayleigh quotient methods for total least squares problems.
In *Recent Advances in Total Least Squares Techniques and Errors–in–Variables Modeling : Proceedings of the Second International Workshop on Total Least Squares and Errors–in–Variables Modeling*, pages 149–160.

Björck, Å., Elfving, T., and Strakoš, Z. (1998).
Stability of conjugate gradient and lanczos methods for linear least squares problems.
SIAM Journal on Matrix Analysis and Applications, 19(3) :720–736.

Björck, Å., Heggernes, P., and Matstoms, P. (2000).
Methods for large scale total least squares problems.
SIAM journal on matrix analysis and applications, 22(2) :413–429.

Carson, E., Higham, N. J., and Pranesh, S. (2020).
Three-precision gmres-based iterative refinement for least squares problems.
SIAM Journal on Scientific Computing, 42(6) :A4063–A4083.

Demmel, J. (1989).
On floating point errors in Cholesky.
University of Tennessee. Computer Science Department.

Demmel, J., Hida, Y., Riedy, E. J., and Li, X. S. (2009).
Extra-precise iterative refinement for overdetermined least squares problems.
ACM Transactions on Mathematical Software (TOMS), 35(4) :1–32.

Freitag, M. A. and Spence, A. (2007).
Convergence theory for inexact inverse iteration applied to the generalised nonsymmetric eigenproblem.
Electronic Transactions on Numerical Analysis, 28 :40–64.

Golub, G. H. and Van Loan, C. F. (1980).
An analysis of the total least squares problem.
SIAM journal on numerical analysis, 17(6) :883–893.

Higham, N. J. and Mary, T. (2022).
Mixed precision algorithms in numerical linear algebra.

Acta Numerica, 31 :347–414.

Higham, N. J. and Pranesh, S. (2021).
Exploiting lower precision arithmetic in solving symmetric positive definite linear systems and least squares problems.
SIAM Journal on Scientific Computing, 43(1) :A258–A277.

Hnětynková, I., Plesinger, M., and Strakoš, Z. (2013).
The core problem within a linear approximation problem $ax \approx b$ with multiple right-hand sides.
SIAM Journal on Matrix Analysis and Applications, 34(3) :917–931.

Kelley, C. (2022).
Newton's method in mixed precision.
SIAM Review, 64(1) :191–211.

Simoncini, V. and Eldén, L. (2002).
Inexact rayleigh quotient-type methods for eigenvalue computations.
BIT Numerical Mathematics, 42(1) :159–182.

Tisseur, F. (2001).
Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems.
SIAM Journal on Matrix Analysis and Applications, 22(4) :1038–1057.

Van Huffel, S. (1991).

Iterative algorithms for computing the singular subspace of a matrix associated with its smallest singular values.

Linear algebra and its applications, 154 :675–709.

Van Huffel, S. and Vandewalle, J. (1991).

The total least squares problem : computational aspects and analysis.

SIAM.

Wilkinson, J. H. (1963).

Rounding errors in algebraic processes.

Prentice-Hall.

Yamazaki, I., Tomov, S., and Dongarra, J. (2015).

Mixed-precision cholesky qr factorization and its case studies on multicore cpu with multiple gpus.

SIAM Journal on Scientific Computing, 37(3) :C307–C330.

More examples - δ matrix, $\delta = 10^{-2}$

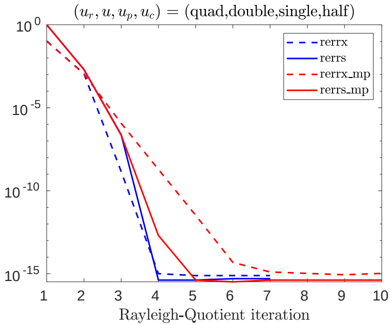
$(A + E)x = b + e,$

$E = 10^{-8}\bar{E} \cdot A, \quad e = 10^{-8}\bar{e} \cdot b$

$\bar{E}$ - random matrix,

$\bar{e}$ - random vector

$$\begin{bmatrix} \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \delta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$



More examples - Random matrix, $m = 100$, $n = 60$

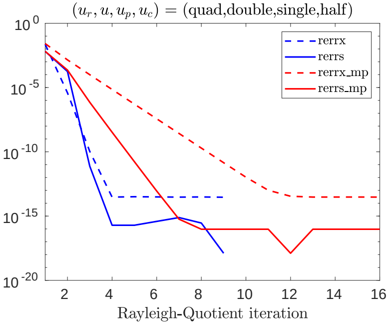
$(A + E)x = b + e$, where

$A \in \mathbb{R}^{m \times n}$ - random matrix,

$b \in \mathbb{R}^{m}$ - vectors of ones,

$E = 10^{-6}\bar{E}, \quad e = 10^{-6}\bar{e}$

$\bar{E}$ - random matrix, $\bar{e}$ - random vector



More examples - [Björck et al., 1998], $m = 30$, $n = 15$

Used to test the convergence properties of RQI-PCGTLS

$(A + E)x = b + e$, where

$E = 0.05\bar{E}, \quad e = 0.05\bar{e}$

$\bar{E}$ - random matrix, $\bar{e}$ - random vector

$b = Ax$ with $x = (1, 1/2, \dots, 1/n)$,

$A = Y \begin{bmatrix} D \\ 0 \end{bmatrix} Z^T \in \mathbb{R}^{m \times n}$, where

Y, Z - random orthogonal matrices

$D = \mathrm{diag}(1, 2^{-1}, \dots, 2^{-n+1})$