Classical Gram-Schmidt without Reorthogonalization

Mark Arnold

Department of Mathematical Sciences, University of Arkansas, Fayetteville

May 5, 2022

Classical Gram-Schmidt: $[Q, R] = \text{CGS}(A)$
Input: $A = [a_1, a_2, ..., a_n] \in \mathbb{R}^{m \times n}$, full column rank
Output: $Q, R : A = QR, \quad Q^T Q = I, \quad R$ upper triangular

$\rho = ||a_1||_2$
$Q = [a_1/\rho], \quad R = [\rho]$
For j=2:n
$r = Q^T a_j$
$s = a_j - Qr$
$\rho = \|s\|_2$
$Q = [Q \;\; s/\rho], \quad R = \begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix}$
EndFor

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

In exact arithmetic:

$$A = \mathcal{Q}\mathcal{R}, \quad \mathcal{Q} \in \mathbb{R}^{m \times n}, \quad \mathcal{Q}^{\mathcal{T}}\mathcal{Q} = I, \quad \mathcal{R} = [\nwarrow\!\!\searrow] \in \mathbb{R}^{n \times n}$$

In finite precision:

$$A = QR + F, \quad R = [\nwarrow\!\!\searrow]$$

- Q and R are the *computed* factors,
- The residual F is small (for CGS $\|F\|_2 \leq (2^{3/2}mn + 2n^{1/2})\|A\|_2\mathbf{u}$).

Theorem: If the (thin) QR factorization $A = QR$ is computed with CGS and R is column-diagonally dominant, then

$$\|I - Q^T Q\|_F \le 3mn^{3/2}\kappa_F(R)\mathbf{u} + \mathrm{O}(\mathbf{u}^2).$$

Proof: sketched in next few slides...

In the notation of the CGS algorithm above, R is "column-diagonally dominant" iff $\|r\|_1 \le \rho$ at each step.

$\kappa_F(R) \equiv \|R\|_F \|R^{-1}\|_F$

Q is *numerically* invariant under column scaling of A:

If $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, $d_{1:n} \neq 0$, and $AD = QU$ is a QR factorization, then

$$A = Q(UD^{-1}) \equiv QR \text{ is a QR factorization of} A.$$

To simplify discussion of the error analysis*, we will assume $\|Ae_j\|_2 = 1$.

Now $\|F\|_2 \leq (3mn + 2n^{1/2})\mathbf{u}$ is slightly cleaner than $\|F\|_2 \leq (3mn + 2n^{1/2})\|A\|_2\mathbf{u}$, since $\|A\|_2$ was a simplifying bound on $\|a_j\|_2$.

*The scaling doesn't need to be implemented for the results to hold (e.g., our code uses a base-2 scaling simply to avoid overflow possibilities...).

We are interested in $\|E\| = \|I - Q^T Q\|$, and thus consider

$$R - Q^T A = R - Q^T(QR + F) = (I - Q^T Q)R - Q^T F = ER - Q^T F:$$

For CGS, $\text{sut}(R) \equiv \text{sut}(\text{fl}(Q^T A))$, so

$$\text{sut}(ER) = \text{sut}(Q^T F)$$

We will use ER (actually $R^T E$) to bound $\|E\|$.

This shares some similarity with Giraud's (Giraud et al., 2005) analysis (almost) showing that CGS loss of orthogonality was proportional to $\kappa^2(A)$, and Smoktunowicz's "rescue" (Smoktunowicz et al., 2006); those papers use $R^T ER \approx R^T R - A^T A$ rather than $R^T E$ to bound $\|E\|$.

At end of step $k+1$: $R_{k+1} = \begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix}$, $\|r\|_2 \leq \|r\|_1 \leq \rho \leq 1$,

$$E_{k+1} \equiv I - Q_{k+1}^T Q_{k+1} = \begin{bmatrix} E & (f - Er)/\rho \\ (f - Er)^T/\rho & \rho(1-q^T q) \end{bmatrix},$$

- E and Er/ρ are the price we pay in this step for loss of orthogonality in the previous steps.
- f/ρ and $\rho(1 - q^T q)$ are rounding errors from this step (local error).

$$R_{k+1}^T E_{k+1} = \begin{bmatrix} R^T & -R^T Er/\rho \\ 0 & -r^T Er/\rho \end{bmatrix} + G,$$

$$\|G\|_2 = \left\| \begin{bmatrix} 0 & R^T f/\rho \\ f^T/\rho & r^T f/\rho + \rho(1-q^T q) \end{bmatrix} \right\|_2 \leq \sqrt{2}\|F\|_2.$$

$$R_{k+1}^T E_{k+1} = \left[\begin{array}{cc} R^T & -R^T Er/\rho \\ 0 & -r^T Er/\rho \end{array} \right] + G, \quad \|r\|_1 \leq \rho$$

We can bound the last column of $R_{k+1}^T E_{k+1}$:

- $\|R^T Er/\rho\|_2 \leq \max\limits_{\|r\|_1=1} \|R^T Er\|_2 \equiv \|R^T E\|_{2,1} = \max\limits_j \|R^T Ee_j\|_2$
- $r^T Er/\rho = (\|a\|_2^2 - \|r\|_2^2 - \rho^2 + \text{local error})/\rho$, (recall: $\rho \geq 2^{-1/2}$)

...and thus $\|R_{k+1}^T E_{k+1}\|_{2,1} \leq \|R^T E\|_{2,1} + \text{local error}$
Finally,

$$\begin{array}{rcl} \|E\|_F & = & \|R^{-T} R^T E\|_F \leq \|R^{-T}\|_F \ \|R^T E\|_F \\ & \leq & \|R^{-1}\|_F \ \sqrt{n} \|R^T E\|_{21} \leq (\|R^{-1}\|_F \ \|R\|_F) \ 4mn^{3/2}\mathbf{u} + \mathrm{O}(\mathbf{u}^2) \end{array}$$

Selective Reorthogonalization Criteria

$\rho = ||a_1||_2, \quad Q = [a_1/\rho], \quad R = [\rho]$
For j=2:n
    $r = Q^T a_j, \quad s = a_j - Qr, \quad \rho = \|s\|_2$
    If reort$(\rho, \|r\|, \text{tol})$ %Could be a While
        $u = Q^T s, \quad r = r + u, \quad s = s - Qu, \quad \rho = \|s\|_2$
    End
    $Q = [Q \;\; s/\rho], \quad R = \begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix}$
EndFor

Historically, reort$(\rho, \|r\|, \text{tol})$ has looked like

$$\|r\|_2 > K\rho, \text{ with } tol = K = 2^{-1} \text{ or } 2^{-1/2} \text{ or } 1 \text{ or } 2 \text{ or } 10 \text{ or } \ldots$$

More recently (2003, Giraud & Langou (in MGS context)) suggested

$$\|r\|_1 > L\rho, \text{ with } L < 1.$$

N.B. $\|r\|_1 \leq \rho$ is column-diagonal dominance in $R\ldots$

...But don't reorthogonalize:

The next few algorithms will have a structure like CGS with selective reorthogonalization. Rather than reorthogonalize, we use Giraud's $L-$criterion to partition.

This can be set in Carson's (Carson et al., 2022) Block Gram-Schmidt (BGS) framework, but with variable block sizes. In their language intra-ortho ("muscle") is CGS, and inter-ortho ("skeleton") is BGS.

The methods we present are BLAS2 versions, not taking advantage of BLAS3 speedup nor flexibility. For example, the BGS inter-ortho consists of matrix-matrix products, while the methods we will present compute these matrix-matrix products as a collection of matrix-vector multiplies.

$[Q, R, P] = \text{CGSReject}(A)$
Input: $A = [a_1, a_2, ..., a_n] \in \mathbb{R}^{m\times n}$, full column rank
Output: $Q, R, P : AP = [Q_1 R_1,\ Q_1 R_2 + Q_2],\ Q_1^T Q_1 = I,\ \ R_1$ upper, $\ \ P$ perm

$\rho = ||a_1||_2, \quad Q_1 = [a_1/\rho], \quad R_1 = [\rho]$
For j=2:n

$r = Q^T a_j, \quad s = a_j - Qr, \quad \rho = \|s\|_2$

If $\|r\|_1 \le L\rho$,

$$Q_1 = [Q_1 \ \ s/\rho], \quad R_1 = \begin{bmatrix} R_1 & r \\ 0 & \rho \end{bmatrix} \text{(accepted)}$$

Else,

$$Q_2 = [Q_2 \ \ s], \quad R_2 = \begin{bmatrix} R_2 & r \\ 0 & 1 \end{bmatrix} \text{(rejected)}$$

End

Update rejected columns (Q_2 and R_2) by those subsequently accepted*

EndFor

% Adding recursion below would give $AP = QR$, R “psychologically” triangular
%If any rejected,
% $\quad [Qnew, Rnew, Pnew] = \text{CGSReject}(Q_2)$
% $\quad$ Update Q, R, P with $Qnew, Rnew, Pnew$
%End

$$AP = QR$$

R "psychologically upper triangular" (old term): Just means that R is permutation equivalent to a triangular matrix.
In fact $AP = QR$ gives the QR-factorization of A as

$$A = (QP)\,(P^T RP).$$

This means, e.g. that we can use CGSReject as an Intra-Ortho BGS method.

Update rejected columns (Q_2 and R_2) by those subsequently accepted

After a pass of CGSReject the columns of Q have been partitioned as

$$Q_1 = [q_{a_1}, q_{a_2}, \ldots, q_{a_{n_a}}] \text{ and } Q_2 = [q_{r_1}, q_{r_2}, \ldots, q_{r_{n_r}}]$$

Each q_{r_i} has associated with it a (possibly empty) set of q_{a_j} for which it has "missed" the orthogonalization, namely all q_{r_i} for which $r_i < a_j$. Name these $q_{a_1}, \ldots, q_{a_{j(i)}}$.

There are at least 2 obvious ways to update:

1. CGS update:
   For i=$r_1, \ldots, r_{n_r}$,
   $V = [q_{a_1}, \ldots, q_{a_{j(i)}}], \quad rr = V' * q_{r_i}, \quad q_{r_i} = q_{r_i} - V * rr,$
   $R(r_1, \ldots, b_{r_{j(i)}}, b_i) = rr$
   End
2. MGS (row) update:
   For j=$a_1, \ldots, a_{n_a}$,
   $V = [q_{r_1}, \ldots, q_{r_{i(j)}}], \quad rr = r(j, r_1, \ldots, r_{j(i)})$
   $rr = q'_{a_j} * V, \quad [q_{r_1}, \ldots, q_{r_{i(j)}}] = V - q_{a_j} rr$
   End

C2MGS

$$A = [Q_1, Q_2, \ldots, Q_K] \begin{bmatrix} R_{11} & R_{12} & \ldots & R_{1K} \\ & R_{22} & \ldots & R_{2K} \\ 0 & & \ddots & R_{KK} \end{bmatrix}$$

$Q_j \in \mathbb{R}^{m \times n_j}$, $R_{ij} \in \mathbb{R}^{n_i \times n_j}$, and n_j, $j = 1{:}K$, are determined at runtime by L-criterion

- $L \to 0$: pure MGS (row-implementation)
- $L \to \infty$: pure CGS
- Analysis *not* done, but we believe that with $L = 1$, this would be a backward stable orthogonalization for Krylov methods, like, e.g. Arnoldi or GMRes

$[Q, R, Blks] = \text{C2MGS}(A)$

Input: $A = [a_1, a_2, ..., a_n] \in \mathbb{R}^{m \times n}$, full column rank

Output: $Q, R:\quad A = QR, \quad Q^TQ = I, \quad R$ upper, $\quad Blks$: indices into blocks

$r_{11} = ||a_1||_2, \quad V = [a_1/r_{11}], \quad R_1 = [r_{11}], \quad K = 1, \quad CurBlkSiz = 1$

For j=2:n

    $w = a_j$

    For k = 1:K-1 (Block MGS on current column)

        $r(Blks_k) = Q_k^T w, \quad w = w - Q_k r$

    End

    $r(CurBlk) = V^T w, \quad s = w - Vw, \quad r_{jj} = \rho = \|w\|_2$ (CGS)

    If $\|r\|_1 \leq L\rho$,

        $++CurBlkSiz, \quad CurBlk = [CurBlk \;\; j]$ (grow current block)

        $V = [V \;\; s/\rho]$

    Else,

        $++K, \quad Blks_K = CurBlk$ (birth new block)

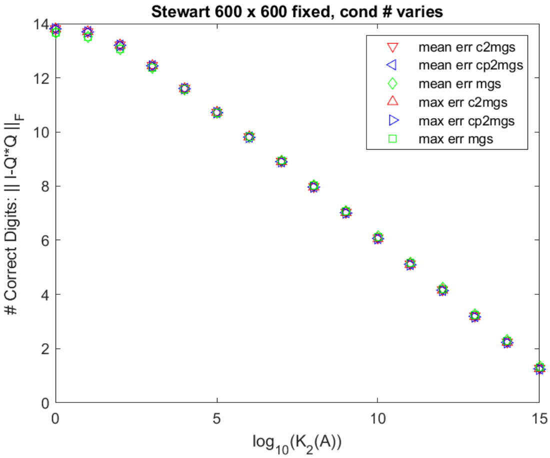
        $Q_K = V, \quad V = [w]$

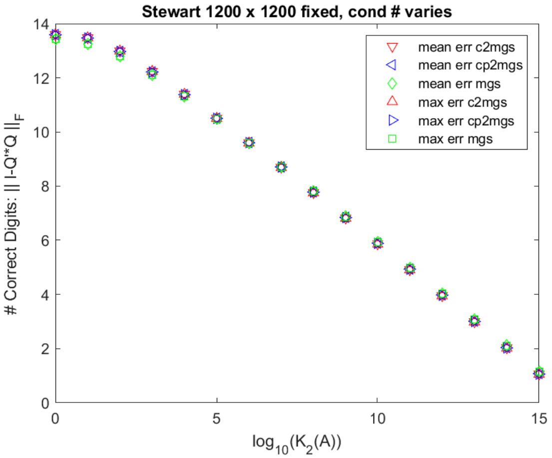
    End

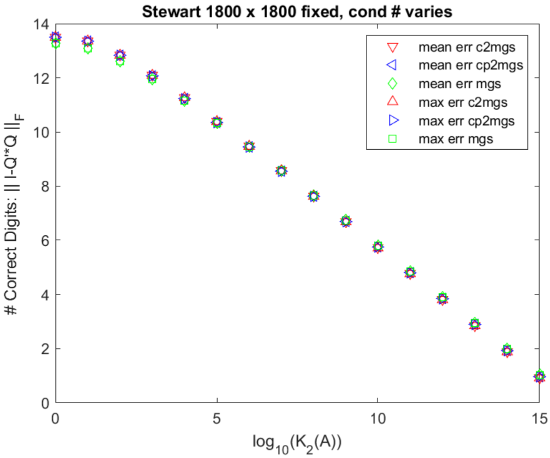
EndFor

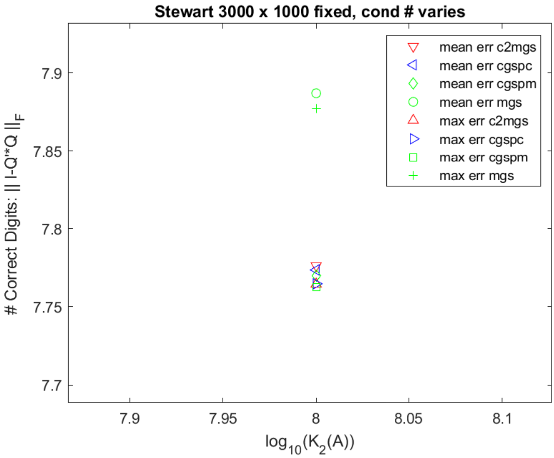
All computational examples presented here were done on my desktop (Dell Precision 2480, Matlab 2022a) or the Mac laptop of Caroline Jennings.

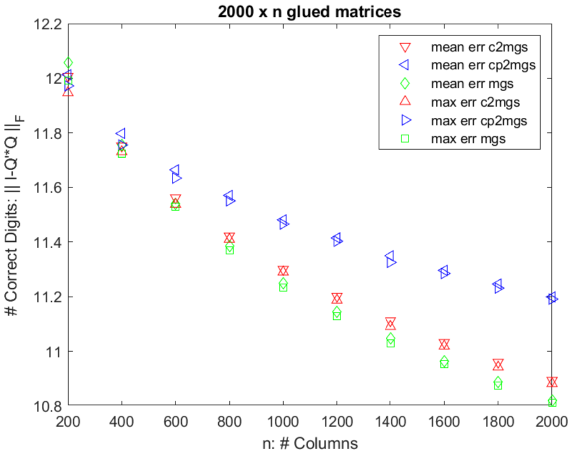
Caroline is graduating in the next few days with Honors and with majors in Mathematics and Creative Writing. She has been working with CGS/MGS just for fun!

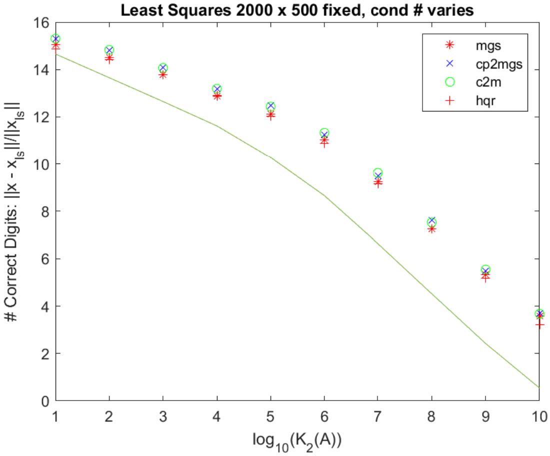

Stewart 600 x 600 fixed, cond # varies
mean err c2mgs
mean err cp2mgs
mean err mgs
max err c2mgs
max err cp2mgs
max err mgs
# Correct Digits: || I-Q'*Q ||F
log10(K2(A))
0
2
4
6
8
10
12
14
0
5
10
15


Stewart 1200 x 1200 fixed, cond # varies
# Correct Digits: || I-Q*Q ||F
log10(K2(A))
mean err c2mgs
mean err cp2mgs
mean err mgs
max err c2mgs
max err cp2mgs
max err mgs
0
2
4
6
8
10
12
14
0
5
10
15


Stewart 1800 x 1800 fixed, cond # varies
# Correct Digits: || I-Q*Q ||F
log10(K2(A))
mean err c2mgs
mean err cp2mgs
mean err mgs
max err c2mgs
max err cp2mgs
max err mgs
0
2
4
6
8
10
12
14
0
5
10
15


Stewart 3000 x 1000 fixed, cond # varies
# Correct Digits: || I-Q'*Q ||F
log10(K2(A))
7.9
7.85
7.8
7.75
7.7
7.9
7.95
8
8.05
8.1
mean err c2mgs
mean err cgspc
mean err cgspm
mean err mgs
max err c2mgs
max err cgspc
max err cgspm
max err mgs


2000 x n glued matrices
Correct Digits: || I-Q'*Q ||F
n: # Columns
mean err c2mgs
mean err cp2mgs
mean err mgs
max err c2mgs
max err cp2mgs
max err mgs
12.2
12
11.8
11.6
11.4
11.2
11
10.8
200
400
600
800
1000
1200
1400
1600
1800
2000


Least Squares 2000 x 500 fixed, cond # varies
# Correct Digits: ||x - x_ls||/||x_ls||
log10(K2(A))
mgs
cp2mgs
c2m
hqr
0
2
4
6
8
10
12
14
16
1
2
3
4
5
6
7
8
9
10