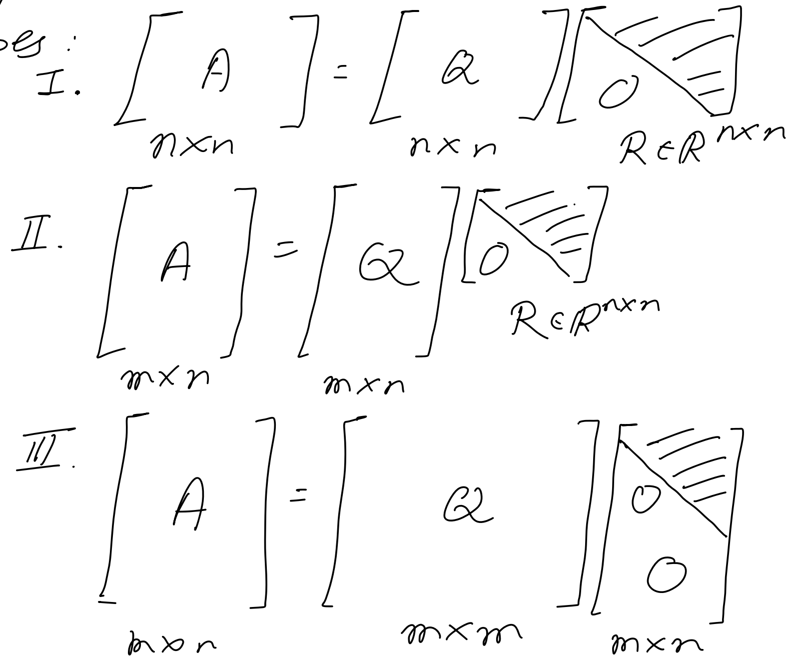
QR Decomposition

\# Recall $Q \in \mathbb{R}^{n\times n}$ is orthogonal if $Q^TQ = QQ^T = I_n$.

\# Given $A \in \mathbb{R}^{m\times n}$, $m \geq n$, we would like to find an orthogonal Q and an upper triangular R s.t. $A = QR$

\# Shapes:



Q. Why is QR important? $\rightarrow$ can be used to solve the least sq. problem:

$$\min_x \|Ax - b\|_2$$

Let $f(x) = (Ax-b)^T(Ax-b) = x^TA^TAx - 2x^TA^Tb + b^Tb$

$\frac{\partial f}{\partial x} = 0$ yields $2A^TAx - 2A^Tb = 0$

or $A^TAx = A^Tb \rightarrow$ <u>normal eqns</u>

\# Can be solved using Cholesky decomp if A is full rank

[Since then A^TA is SPD]

\# However sensitivity can be high for

ill-conditioned A. Recall $\kappa(A^TA) = [\kappa(A)]^2$

\# Let $A = QR$. Then the normal eqns become

$$\underbrace{R^TQ^TQ}_{I}Rx = R^TQ^Tb \quad \left[\begin{matrix}\text{If } A \text{ is full rank} \\ R \text{ is full rank}\end{matrix}\right]$$

$$\Rightarrow \quad Rx = Q^Tb$$

upper triangular $\Longrightarrow$ Use backward substitution

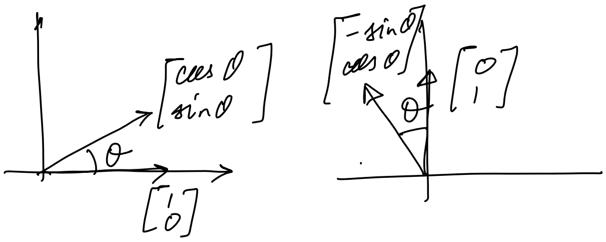
The nature of Q

\# Clearly $\underbrace{\|Qx\|_2 = \|x\|_2}_{\text{2-Norm preserving}}$ $\left[\begin{matrix}\because (Qx)^TQx \\ = x^TQ^TQx = x^Tx\end{matrix}\right]$

Rotation

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

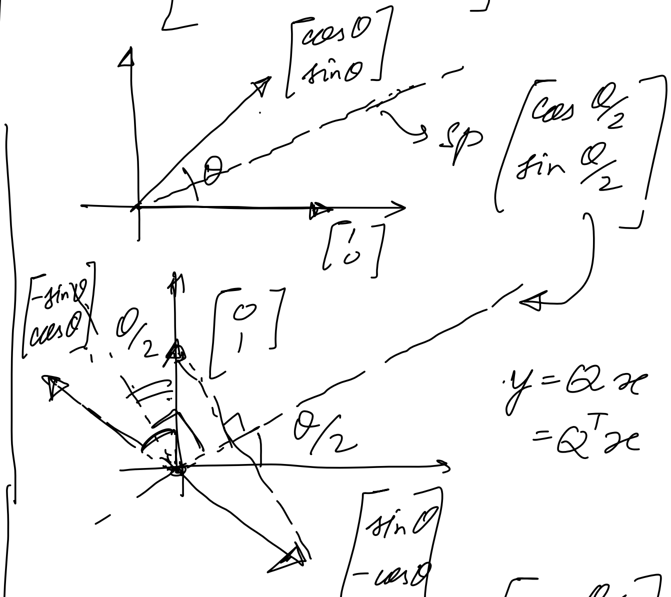
\# Consider action on $\begin{bmatrix}1\\0\end{bmatrix}$ & $\begin{bmatrix}0\\1\end{bmatrix}$



$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Reflection

$$Q = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} = Q^T$$



$$y = Qx = Q^Tx$$

\# Reflection about span $\begin{bmatrix}\cos\theta/2 \\ \sin\theta/2\end{bmatrix}$

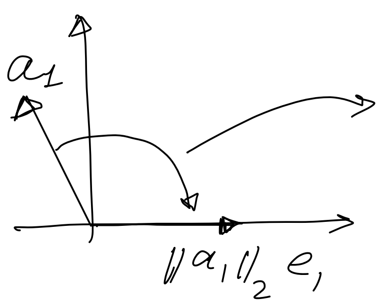
Q. If $A = QR \Rightarrow Q^T A = R$. Then can we design Q so that Q^T acts like L^{-1} to upper triangularize A to R.

Similar to LU: we want

$$\begin{bmatrix} Q_1^T \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ a_1 & a_2 & \cdots & a_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} = \begin{bmatrix} * & * & \cdots & * \\ 0 & & & \\ \vdots & & * & \\ 0 & & & \end{bmatrix}$$

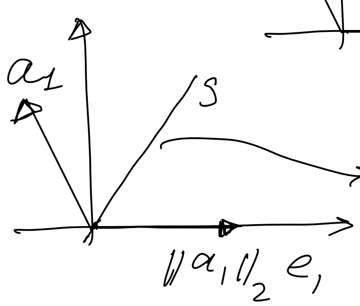
$\Rightarrow \quad Q_1^T a_1 = \pm \|a_1\|_2 \, e_1 \quad \leftarrow$ [Since we already know $\|Q_1^T a_1\|_2 = \|a_1\|_2$]

We want



Can be done with either reflection or rotation.

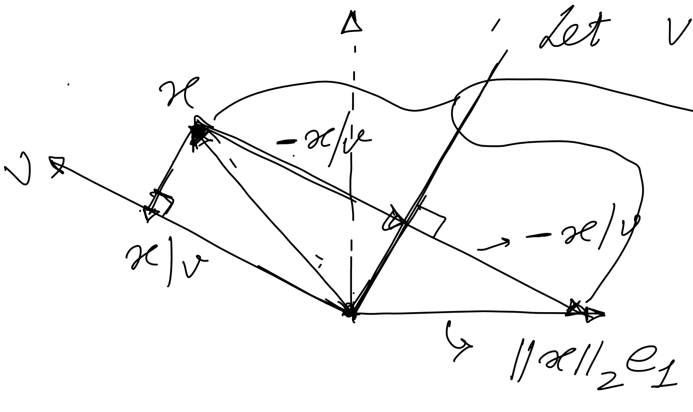
→ Choose reflection first.



Have to find this line S.

Initial idea through 2-D geometry

Let v be a line $\perp$ to S.



$\rightarrow -2x|v$

So $\underline{x - 2x|v = \|x\|_2 e_1}$

Suggest a method to v. Q

\# $x|v = v^T x \cdot \frac{v}{\|v\|_2^2} = v^T x \cdot \frac{v}{v^T v}$

Then $x - 2\frac{v^T x \cdot v}{v^T v} = x - \frac{2vv^T}{v^T v}\cdot x$

$$= \underbrace{\left[I - \frac{2vv^T}{v^T v}\right]}_{} x$$

$P :=$ Reflection Matrix about the plane orthogonal to v.

\# $P = I - \beta vv^T$; $\beta = \frac{2}{v^T v}$.

\# Check: $Pv = -v$ & $Px = x$ if $v^T x = 0$.

$$\left[I - \beta vv^T\right] v = v - 2\frac{vv^T v}{v^T v} = -v$$

$$\Rightarrow Px = x - \frac{2vv^T x}{v^T} = x$$

$$P^T P = \left(I - \frac{2vv^T}{v^Tv}\right)\left(I - \frac{2vv^T}{v^Tv}\right) = I - 4\frac{vv^T}{v^Tv} + 4\frac{vv^T}{v^Tv} = I$$

Q. How to choose v?

\# (*) can be satisfied by choosing

$$v = x \pm \|x\|_2 e_1$$

Check: $Px = \left[I - \frac{2vv^T}{v^Tv}\right] x$

$$= x - 2\frac{v^T x}{v^T v}\cdot v$$

$$= x - 1 . v$$

$$= x - \left[x \pm \|x\|_2 e_1\right] = \mp \|x\|_2 e_1$$

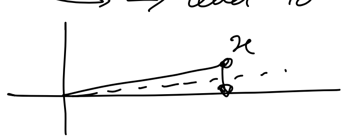
$$v^T x = x^T x \pm \|x\|_2 x_1 = \|x\|_2^2 \pm \|x\|_2 x_1$$

$$v^T v = x^T x \pm 2\|x\|_2 x_1 + \|x\|_2^2 = 2\left[\|x\|_2^2 \pm \|x\|_2 x_1\right]$$

Computing v: $v_1 = x_1 - \|x\|$

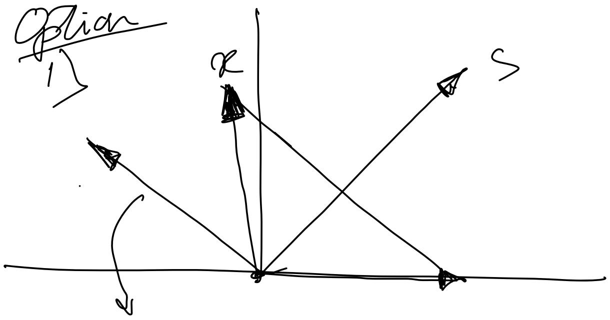
$$\begin{bmatrix} v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \vdots \\ x_n \end{bmatrix}$$

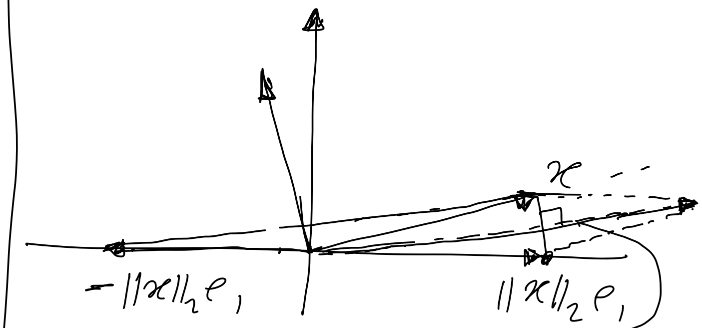
However if $x_1 > 0$ & $x_2^2 + \cdots + x_n^2 \ll x_1^2$ $\Rightarrow$ lead to large errors



Then there are two options:

Option 1





$v = x - \|x\|_2 e_1$

If $x_1 < 0$, choose this v.

$v = x + \|x\|_2 e_1$

If $x_1 > 0$, choose this v.

Option 2: Choose $v = x - \|x\|_2 e_1$ in both cases, but calculate it differently.

If $x_1 < 0$, $v_1 = x_1 - \|x\|_2$

If $x_1 > 0$, $v_1 = \dfrac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \|x\|_2}$

$$\left[\; x_1 - \|x\|_2 = \frac{(x_1 - \|x\|_2)(x_1 + \|x\|_2)}{(x_1 + \|x\|_2)} = \frac{x_1^2 - \|x\|_2^2}{x_1 + \|x\|_2} \;\right]$$

Householder Reflection: maps x to $\|x\|_2 e_1$ is given by $P = I - \beta v v^T$ where $v = x \pm \|x\|_2 e_1$, $\beta = \dfrac{2}{v^T v}$

Iterate:

$$Q_{n-1}^T \cdots Q_1^T A = R$$

or $A = Q_1 \cdots Q_{n-1} R = QR = \underbrace{\begin{bmatrix} Q \end{bmatrix}}_{m \times n} \underbrace{\begin{bmatrix} \diagdown \\ 0 \\ 0 \end{bmatrix}}_{R}$

Here $Q_j = I_n - \beta_j v^{(j)} [v^{(j)}]^T$ $\quad m > n.$

Q_j's are not formed explicitly. $A \in \mathbb{R}^{m\times n}$

Reasons:

1) $QA = (I - \beta v v^T)A = A - (\beta v)\underbrace{(v^T A)}_{2mn}$

$2mn \rightarrow$ matrix-vector mult.

$+\,mn \rightarrow$ rank one update

$m\times n$ matrix subtraction $\rightarrow mn$

$= 4mn$ flops

Same for AQ.

Computation of v, β by option 2 takes $3m$ flops (Exercise)

Whereas QA would take $O(m^2 n)$

Hence the typical steps are:

Normalize $v = \dfrac{v}{v(1)}$ s.t. $v(1) = 1$. (don't have to store it)

for $j = 1:n$
 $[v, \beta] = \text{householder}(A[j:m, j])$ $\rightarrow 3m$
 $A[j:m, j:n] = A[j:m, j:n] - \beta v (v^T A[j:m, j:n])$ $\rightarrow 4(m-j)(n-j)$
 $A[j+1:m, j] = v[2:m-j+1]$

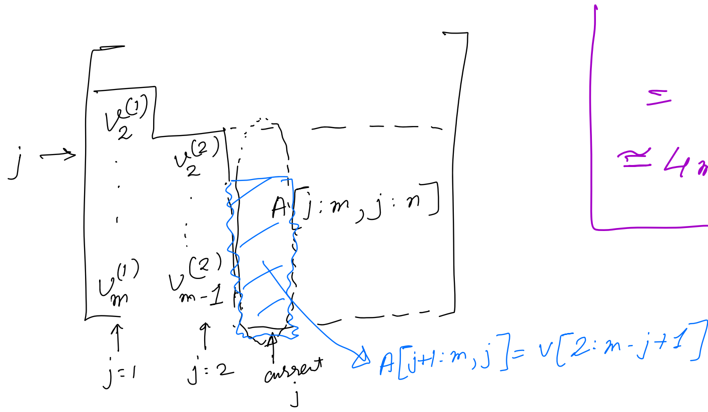
$v(1) = 1$

$$3mn + 4mn^2 - \frac{4m\,n(n+1)}{2} - \frac{4n^2(n+1)}{2} + 4\left[\frac{n^3}{6} + \dots\right]$$

$$= \quad \simeq 4mn^2 - 2mn^2 - 2n^3 + \frac{2}{3}n^3$$

$$\simeq 2mn^2 - \frac{4}{3}n^3$$

$$= 2n^2\left[m - \frac{n}{3}\right]$$



Resulting Q & R

This method of storing $v^{(j)}$'s instead of Q or Q_j's is called "Factored-form" Representation.

2) If (say) we need to compute $Q^T C$ then we execute:

for $j = 1:n$
  $C = Q_j C$
end

This is called a factored form representation of Q

3) Q can be calculated on demand

$Q = I_m$
for $j = 1:n$
  $Q = Q Q_j$
end

Forward accumulation

Flops (pg 238 Golub)
$4mnk - 2(m+k)n^2 + \frac{4}{3}n^3$

or

$Q = I_m$
for $j = n:-1:1$
  $Q = Q_j Q$
end

Backward accumulation (better — why?)
(initially Q is mostly I_m)

QR (Householder) is backward stable (without proof)

If computed v is denoted by $\hat{v}$, &
$\hat{P} = \left[I - 2\frac{\hat{v}\hat{v}^T}{\hat{v}^T\hat{v}} \right]$ then $\|\hat{P} - P\|_2 = O(u) \rightarrow$ forward error in P.

B.E. : $fl(\hat{P}A) = P(A+E)$ $\|E\|_2 = O(u\|A\|_2)$

$fl(A\hat{P}) = (A+E)P$ $\|E\|_2 = O(u\|A\|_2)$

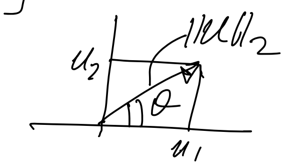
Rotations for QR : Rotations in n-D are harder to set-up (compared to reflections)

But rotations are very easy in 2-D.

e.g. in 2-D, a rotation that rotates $u = [u_1, u_2]^T$ to $\|u\|_2 e_1$ is

$$G_2^T = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$ [Note: $G_2 u$ rotates u clockwise]

where $c = \dfrac{u_1}{\|u\|_2}$, $s = \dfrac{-u_2}{\|u\|_2}$



<u>Givens Rotation</u>: Used to introduce <u>one zero</u> - as opposed to a full column of zeros

$$G_{ik} = \begin{array}{c} \left[\begin{array}{ccccccc} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & & 0 & & 0 & & 1 \end{array}\right] \begin{array}{c} \\ \\ \to i \\ \\ \to k \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} \quad & & \uparrow i & & \uparrow k & & \end{array} \end{array}$$

where $c = \cos\theta$, $s = \sin\theta$ for some θ.

G_{ik} is orthogonal

$G_{ik}^T A$ = counterclockwise rot. by θ radians in the $(i-k)$ plane

If $y = G_{ik}^T x$, $y_j = \begin{cases} c x_i - s x_k & j = i \\ s x_i + c x_k & j = k \\ x_j & j \neq i, k \end{cases}$

Hence we can set y_k to be zero by setting $c = \dfrac{x_i}{\sqrt{x_i^2 + x_k^2}}$, $s = \dfrac{-x_k}{\sqrt{x_i^2 + x_k^2}}$

Suppose we want to apply Given's rotation $A = G_{ik}^T A$, <u>we</u> need to update only

$$A[[i,k],:] = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A[[i,k],:] \qquad \underline{A \in \mathbb{R}^{m\times n}}$$

$$\underline{6n \text{ flops}} \leftarrow \underbrace{[\ \ ]}_{2\times 2}\underbrace{[\ \ \ \ \ \ \ \ \ \ ]}_{2\times n}$$

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

$[c, s] = \text{givens}(a, b)$ [say $b \neq 0$, otherwise $c = 1, s = 0$]

if $|b| > |a|$ $\longrightarrow$ [Numerical stability by making $|\tau| < 1$]

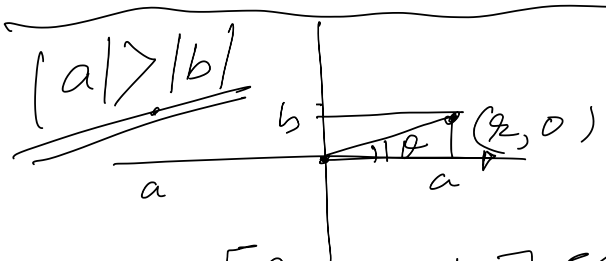
$$\tau = -\frac{a}{b}, \quad s = \frac{1}{\sqrt{1+\tau^2}}; \quad c = s\tau$$

else

$$\tau = -\frac{b}{a}; \quad c = \frac{1}{\sqrt{1+\tau^2}}; \quad s = c\tau$$

end

$\underline{|a| > |b|}$



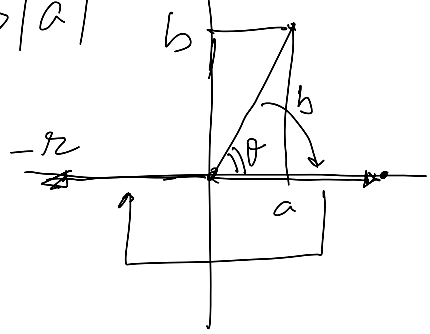
$|a| > |b|$

$$\cos\theta = \frac{1}{\sqrt{1+\frac{b^2}{a^2}}} = \frac{a}{\sqrt{a^2+b^2}}$$

$$\sin\theta = \frac{-b}{\sqrt{a^2+b^2}}$$

$$\underbrace{\begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} & \frac{b}{\sqrt{a^2+b^2}} \\ \frac{-b}{\sqrt{a^2+b^2}} & \frac{a}{\sqrt{a^2+b^2}} \end{bmatrix}}_{\text{clockwise rotation by } \theta} \begin{bmatrix} a \\ b \end{bmatrix} = \underbrace{\begin{bmatrix} +\sqrt{a^2+b^2} \\ 0 \end{bmatrix}}_{r}$$

$|b| > |a|$

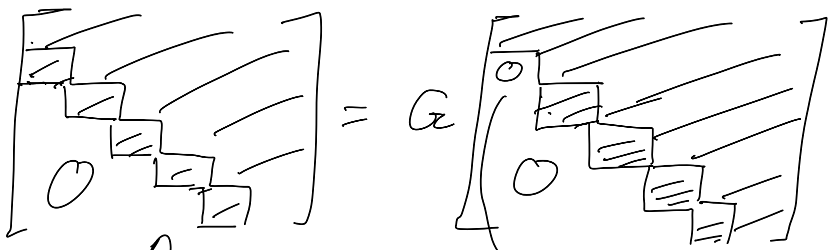


$$\sin\theta = \frac{b}{\sqrt{a^2+b^2}}, \quad \cos\theta = \frac{-a}{\sqrt{a^2+b^2}}$$

$$-\begin{bmatrix} +\frac{a}{\sqrt{a^2+b^2}} & +\frac{b}{\sqrt{a^2+b^2}} \\ \frac{-b}{\sqrt{a^2+b^2}} & +\frac{a}{\sqrt{a^2+b^2}} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \underbrace{\begin{bmatrix} -\sqrt{a^2+b^2} \\ 0 \end{bmatrix}}_{-r}$$

Givens QR Example : Hessenberg QR

e.g.



A

Upper Hessenberg form

zero introduced by Givens rotation.

\# Let $A \in \mathbb{R}^{n \times n}$ be upper Hessenberg. We compute $\underbrace{[G_1 \cdots G_{n-1}]^T}_{\text{Givens rotations}} A = R \hookrightarrow$ upper triangular

for $j = 1 : n-1$

$[c, s] = \text{givens}(A[j,j], A[j+1, j])$

$A[j:j+1, j:n] = \underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A(j:j+1, j:n)}$

end

Flops $= \sum_{j=1}^{n-1} 6(n-j) \approx 6n^2 - \frac{6n^2}{2} \approx 3n^2$

\# A Given's rotation is Backward Stable

$fl[G_{ik}^T A] = G_{ik}^T (A + E)$, $\|E\|_2 = O(u \|A\|_2)$

(proof not included)

\# Sequences of orthogonal update using Givens / Householder matrices are backward stable

i.e.

let $A_k = fl[\hat{Q}_k A_{k-1} \hat{Z}_k]$ $k = 1:p$

$\to$ Householder / Givens matrices generated in a comp.

FACT: $B = \underbrace{(Q_p \cdots Q_1)}_{\substack{\text{exact} \\ H/G \text{ matrices}}}(A+E)\underbrace{(Z_1 \cdots Z_p)}_{\substack{\text{exact} \\ \text{Householder/} \\ \text{Givens} \\ \text{matrices}}}$

with $\|E\| \leq c\, u \|A\|_2$

$\hookrightarrow$ c depends mildly on n, m, p

QR Factorization

Thm: If $A \in \mathbb{R}^{m \times n}$, then there exists an orthogonal $Q \in \mathbb{R}^{m \times m}$ & an upper triangular $R \in \mathbb{R}^{m \times n}$ s.t. $A = QR$

Proof: Use induction on n. Let $n = 1$, & Q is the Householder matrix s.t. $R = Q^T A$

$\Rightarrow A = QR$ exists.

For general n, partition $A = [A_1 \ \underbrace{v}_{A[:, n]}]$

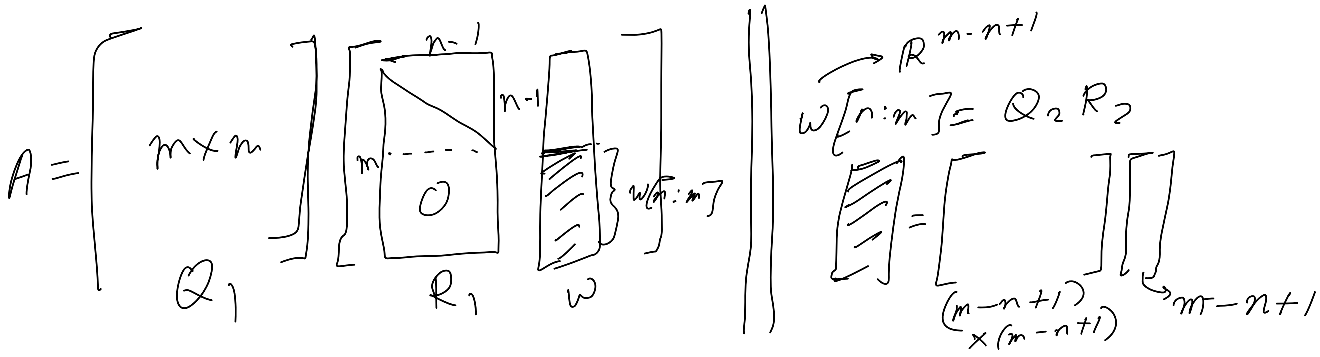
Assume $\exists$ orthogonal $Q_1 \in \mathbb{R}^{m \times m}$ s.t. $R_1 = Q_1^T A_1$ is upper triangular.

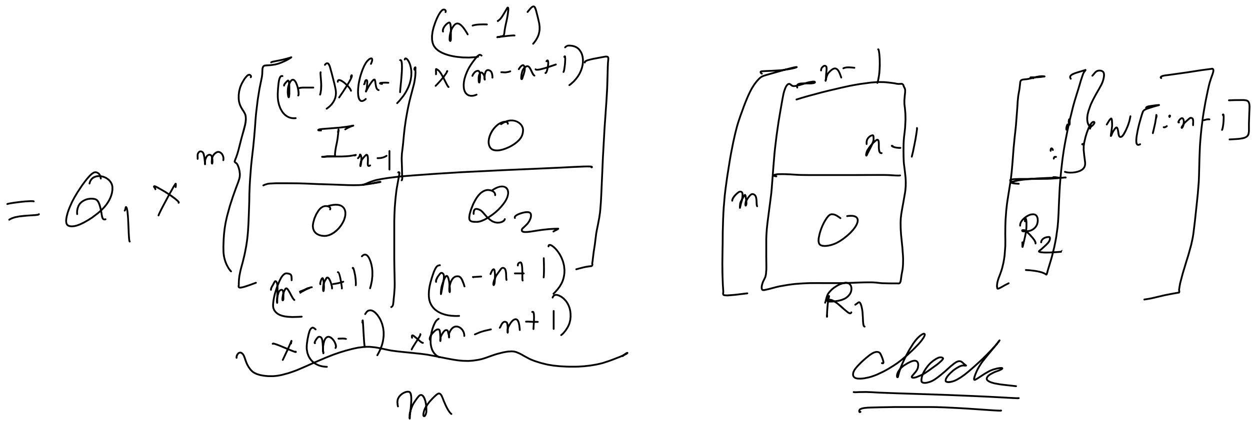
Set $w = Q^T v$ & let $w[n:m] = Q_2 R_2$

$$A = [Q_1 R_1 \quad v] = [Q_1 R_1 \quad Q_1 Q_1^T v]$$
$$= Q_1 [R_1 \quad \underbrace{Q_1^T v}_{w}]$$

Let $w[n:m] = Q_2 R_2$. Then

$$A = \underbrace{Q_1 \begin{bmatrix} I_{n-1} & 0 \\ 0 & Q_2 \end{bmatrix}}_{Q} \underbrace{\left[\begin{array}{c|c} R_1 & \begin{matrix} w[1:n-1] \\ R_2 \end{matrix} \end{array}\right]}_{R}$$





check

<u>Thm:</u> If $A = QR$ is a QR factorization of a full column rank $A \in \mathbb{R}^{m\times n}$ and

$$A = [a_1 | \cdots | a_n]$$
$$Q = [q_1 | \cdots | q_m]$$

1) Then for $k = 1:n$,
$$\text{span}\{a_1, \dots, a_k\} = \text{span}\{q_1, \dots, q_k\}$$
and $r_{kk} \neq 0$.

2) Moreover, if $Q_1 = Q[1:m, 1:n]$, $Q_2 = [1:m, n+1:m]$ & $R_1 = R[1:n, 1:n]$ then
$$\text{range}(A) = \text{range}(Q_1)$$
$$\text{range}(A^{\perp}) = \text{range}(Q_2)$$

3) $A = Q_1 R_1$

Proof: 1) $\begin{bmatrix} \uparrow & & & & \\ a_1 & \cdots & a_k & \cdots & a_n \\ \downarrow & & & & \end{bmatrix} = \begin{bmatrix} q_1 & \cdots & q_m \end{bmatrix} \begin{bmatrix} r_{11} & \cdots & \cdot \\ 0 & \ddots & r_{nn} \\ & 0 & \end{bmatrix}$

$a_{11} = q_1 r_{11}$

$a_2 = q_1 r_{12} + q_2 r_{22}$

$a_k = \sum_{i=1}^{k} r_{ik} q_i \in \text{span}\{q_1, \cdots, q_k\}$

$\Rightarrow \text{span}\{a_1, \cdots, a_k\} \subseteq \text{span}\{q_1, \cdots, q_k\}$

But span $\{a_1, \cdots, a_k\}$ has dim k (A is full col. rank)

$\Rightarrow \text{span}\{a_1, \cdots, a_k\} = \text{span}\{q_1, \cdots, q_k\}$

2,3) $A = QR = \begin{bmatrix} \underbrace{Q_1}_{n} & Q_2 \end{bmatrix} \underbrace{\begin{bmatrix} \overbrace{\diagdown}^{R_1} \\ 0 \end{bmatrix}}_{R} \}n = Q_1 R_1$ (This QR)

Thm: Suppose $A \in \mathbb{R}^{m\times n}$ has full column rank. Then

1) $A = Q_1 R_1$ is unique where $Q_1 \in \mathbb{R}^{m\times n}$ has orthonormal columns and R_1 is upper triangular with +ve diagonal entries.

2) Moreover $R_1 = G^T$ where G is the Cholesky factor of A^TA.

Proof: $A^TA = (Q_1R_1)^T(Q_1R_1) = R_1^TR_1 \rightarrow$ uniqueness of R_1 follows. Then $Q_1 = AR_1^{-1}$ is also unique.

(Classical)
<u>Gram - Schmidt</u>: (Can produce thin Q's directly)

$$m\left\{\underbrace{\begin{bmatrix} | & & \\ a_1 & \cdots & a_n \\ | & & \end{bmatrix}}_{n}\right. = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix} \underbrace{\begin{bmatrix} r_{11} & & \\ & \ddots & \\ 0 & & r_{nn} \end{bmatrix}}_{R}$$

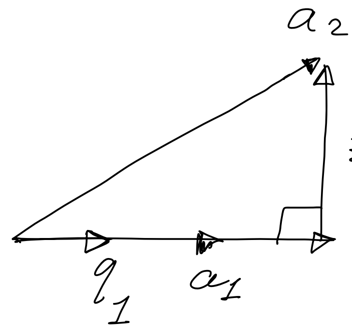
$$a_k = \sum_{i=1}^{k} r_{ik} q_i = r_{kk} q_k + \sum_{i=1}^{k-1} r_{ik} q_i \quad \text{— (1)}$$

$q_1 = \dfrac{a_1}{r_{11}}$. Choose $r_{11} = \|a_1\|_2$

$$z = a_2 - \underbrace{\langle a_2, q_1 \rangle}_{r_{12}} \cdot q_1$$

$$= a_2 - r_{12} q_1$$

$$q_2 = \frac{a_2 - r_{12} q_1}{\|z\|_2}$$



If we know q_j and r_{ij} $\forall\, j<k,\ i \le j$ can we find r_{ik} & q_k?

<u>For r_{ik}</u>: Pre-multiply (1) by q_i^T.

$$r_{ik} = q_i^T a_k \quad \forall\, i = 1, \cdots, k-1$$

<u>For q_k</u>

$$q_k = \left[\frac{a_k - \sum_{i=1}^{k-1} r_{ik} q_i}{r_{kk}}\right] =: \frac{z}{r_{kk}} \quad \text{(known)}$$

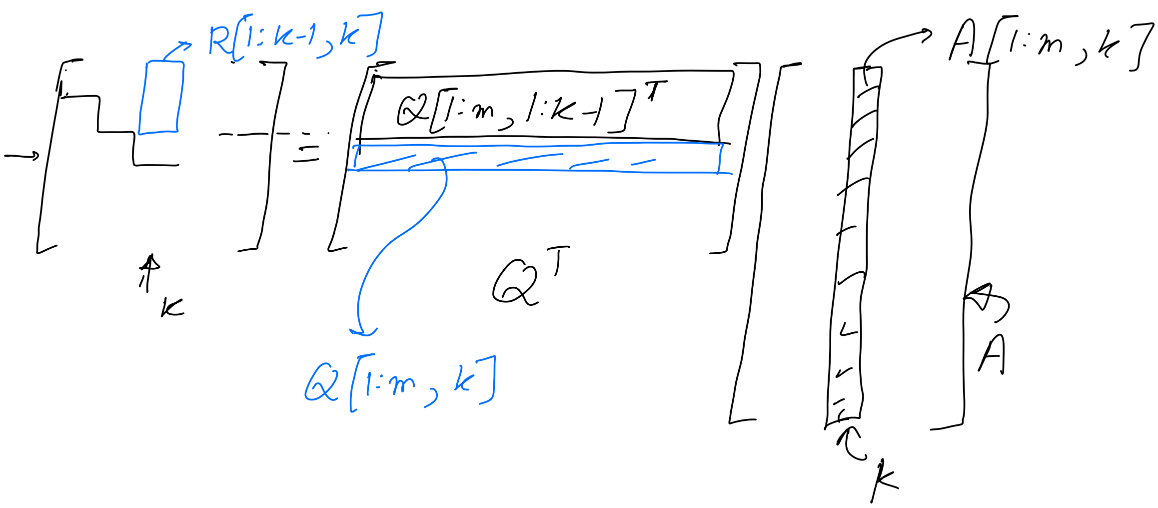
Then $z = q_k r_{kk}$

clearly $r_{kk} = \|z\|_2$

So now, $q_k = \dfrac{z}{r_{kk}}$ can be computed.

Flops = ?
Exercise

$R[1,1] = \|A[:,1]\|_2$
$Q[:,1] = A[:,1]/R[:,1]$
for $k = 2:n$
  $R[1:k-1,k] = Q[1:m,1:k-1]^T A[1:m,k]$
  $z = A[1:m,k] - Q[1:m,1:k-1] \cdot R[1:k-1,k]$
  $R[k,k] = \|z\|_2$
  $Q[1:m,k] = z/R[k,k]$
end



Classical Gram-Schmidt is not Backward Stable

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix} = \underbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & -2^{-1/2} & -2^{-1/2} \\ 0 & 2^{-1/2} & 0 \\ 0 & 0 & 2^{-1/2} \end{bmatrix}}_{\hat{Q}} \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix}}_{\hat{R}}}$$

$\#$ If $\varepsilon^2 \ll u$, then $\hat{Q}$

$$\left.\begin{array}{l} \text{But } \hat{q}_2^T \hat{q}_3 = 1/2 \\ \text{Also } \|\hat{q}_1\| \neq 1 \end{array}\right\} \Rightarrow \hat{Q} \text{ is not orthogonal}$$

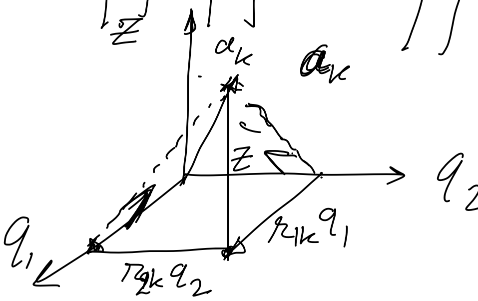
Q. Is this due to poor conditioning of data or due to backward error?

Modified Gram-Schmidt

In classical GS: 1) All r_{ik}, $i = 1, \dots, k-1$ are computed first.

2) Then: $z = a_k - \sum_{i=1}^{k-1} r_{ik} q_i$

$$\underset{z}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} - r_{1k} \begin{bmatrix} q_1 \\ \\ \\ \end{bmatrix} - r_{2k} \begin{bmatrix} q_2 \\ \\ \\ \end{bmatrix} \cdots - r_{k-1,k} \begin{bmatrix} q_{k-1} \\ \\ \\ \end{bmatrix}$$



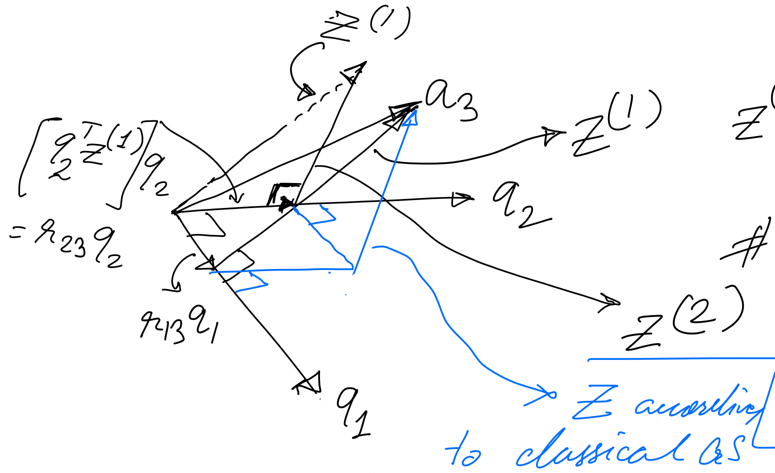
In modified GS: this is done in stages:

1) $r_{13} = q_1^T a_3$ is computed

2) $z^{(1)} = a_3 - r_{13} q_1 \Rightarrow$

3) $r_{23} = q_2^T z^{(1)}$

4) $z^{(2)} = z^{(1)} - r_{23} q_2$



$z^{(2)} \perp \mathrm{sp}\{q_1, q_2\}$

We can then normalize $z^{(2)}$ and use as q_3.

MGS : Given lin. ind. $a_1, \cdots, a_n \to$ calculate $q_1, \cdots, q_n$

Flops

$2m(k-1)$

$2m(k-1)$

$\sum_{k=1}^{n} 4m(k-1)$

$\simeq 4m \frac{n(n+1)}{2}$

$\simeq 2mn^2$

for $k = 1:n$

  for $i = 1, \cdots, k-1$

  $r_{ik} = a_i^T a_k$ → actually q_i

  $z^{(k)}$ $a_k = a_k - r_{ik} a_i$ → the $z^{(k)}$ step above

  end

  $r_{kk} = \|a_k\|_2$ → $z^{(k)}$ in above notation

  $a_k = \frac{a_k}{r_{kk}}$

end → Now q_k is orthonormal

Without roundoff both should produce same output.

But with roundoff $\left[a_k - \sum_{i=1}^{k-1} r_{ik} q_i\right]$ might not be perfectly orthogonal to q_i, $i<k$.

In MGS, since r_{ik}'s are computed at each step, orthogality is enforced repeatedly.

When MGS is applied to $A = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}$,

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & -2^{-1/2} & -6^{-1/2} \\ 0 & 2^{-1/2} & -6^{-1/2} \\ 0 & 0 & \left(\frac{2}{3}\right)^{1/2} \end{bmatrix}$$

$\|q_1\| = \sqrt{1+\varepsilon^2}$ // $|q_1^T q_3| = \frac{\varepsilon}{\sqrt{6}}$

$q_2^T q_3 = 0$

Q) Are the above errors due to the algo or data?

→ need a language for "sensitivity" for rectangular matrices.

→ Orthogonality/Orthonormality is also compromised.

$\|I_m - Q^TQ\|_2$ → measure of deviation from orthonormality.

Condition no of $A \in \mathbb{R}^{m\times n}$ (Assume A - full column rank)

$$\kappa_2(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$$

For Householder generated Q, $\|I_m - Q^TQ\|_2 \approx c u$ (c: small constant)

For MGS → Q, $\|I_m - Q^TQ\|_2 \approx u\, \kappa_2(A)$

<u>Exercise</u>: Compute the flops req. by Householder vs MGS to orthonormalize a set or lin. ind. vectors $\{v_1, \cdots, v_n\}$, $v_i \in \mathbb{R}^m$.