Numerical Linear Algebra
QR and Least Squares

Zahra Lakdawala

October 7, 2021

Outline

Projectors

Projectors

- A projector satisfies $\mathbf{P}^2 = \mathbf{P}$. They are also said to be *idempotent*.
  - Orthogonal projector
  - Oblique projector
- Example

$$\begin{bmatrix} 0 & 0 \\ \alpha & 1 \end{bmatrix}$$

  - is an oblique projector if $\alpha \neq 0$,
  - is orthogonal projector if $\alpha = 0$.

Complementary Projectors

- Complementary projectors: $\mathbf{P}$ vs. $\mathbf{I} - \mathbf{P}$.
- What space does $\mathbf{I} - \mathbf{P}$ project?

Complementary Projectors

- Complementary projectors: $\mathbf{P}$ vs. $\mathbf{I}-\mathbf{P}$.
- What space does $\mathbf{I}-\mathbf{P}$ project?
  - Answer: null($\mathbf{P}$)
  - range($\mathbf{I}-\mathbf{P}$) $\supseteq$ null($\mathbf{P}$) because $\mathbf{Pv}=0 \Rightarrow (\mathbf{I}-\mathbf{P})\mathbf{v}=\mathbf{v}$.
  - range($\mathbf{I}-\mathbf{P}$) $\subseteq$ null($\mathbf{P}$) because for any $\mathbf{v}$

$$(\mathbf{I}-\mathbf{P})\mathbf{v}=\mathbf{v}-\mathbf{Pv}\in \text{null}(\mathbf{P}).$$

- A projector separates $\mathbb{C}^m$ into two complementary subspace: range space and null space (i.e., range($\mathbf{P}$) + null($\mathbf{P}$) $=\mathbb{C}^m$ and range($\mathbf{P}$) $\cap$ null($\mathbf{P}$) $=0$ for projector $\mathbf{P}\in\mathbb{C}^{m\times m}$)
- It projects onto range space along null space
  - In other words, $\mathbf{x}=\mathbf{Px}+\mathbf{r}$, where $\mathbf{r}\in \mathit{null}(\mathbf{P})$
- Question: Are range space and null space of projector orthogonal to each other?

Orthogonal Projector

- An orthogonal projector is one that projects onto a subspace S_1 along a space S_2, where S_1 and S_2 are orthogonal.

Theorem

A projector $\mathbf{P}$ is orthogonal if and only if $\mathbf{P} = \mathbf{P}^*$.

Proof

"If" direction: If $\mathbf{P} = \mathbf{P}^*$, then$(\mathbf{Px})^*(\mathbf{I} - \mathbf{P})\mathbf{y} = \mathbf{x}^*(\mathbf{P} - \mathbf{P}^2)\mathbf{y}$. "Only if" direction: Use SVD. Suppose $\mathbf{P}$ projects onto S_1 along S_2 where $S_1 \perp S_2$, and S_1 has dimension n. Let $q_1, \cdot, q_n$ be orthonormal basis of S_1 and $q_{n+1}, \cdot, q_m$ be a basis for S_2. Let $\mathbf{Q}$ be unitary matrix whose jth column is q_j , and we have $\mathbf{PQ} = (\mathbf{q}_1, \mathbf{q}_2, \cdot \mathbf{q}_n, 0, \cdot, 0)$, so $\mathbf{Q}^*\mathbf{PQ} = diag(1, 1, \cdot, 1, 0, \cdot) = \Sigma$, and $\mathbf{P} = \mathbf{Q}\Sigma\mathbf{Q}^*$.

Question: Are orthogonal projectors orthogonal matrices?

Basis of Projections

- Projection with orthonormal basis
  - Given any matrix $\hat{\mathbf{Q}} \in \mathbb{C}^{m \times n}$ whose columns are orthonormal, then $\mathbf{P} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^*$ is orthogonal projector, so is $\mathbf{I} - \mathbf{P}$
  - We write $\mathbf{I} - \mathbf{P}$ as $\mathbf{P}_\perp$
  - In particular, if $\hat{\mathbf{Q}} = \mathbf{q}$, we write $\mathbf{P}_\mathbf{q} = \mathbf{q}\mathbf{q}^*$ and $\mathbf{P}_{\perp\mathbf{q}} = \mathbf{I} - \mathbf{P}_\mathbf{q}$
  - For arbitrary vector $\mathbf{a}$, we write $\mathbf{P}_\mathbf{a} = \frac{\mathbf{a}\mathbf{a}^*}{\mathbf{a}^*\mathbf{a}}$ and $\mathbf{P}_{\perp\mathbf{a}} = \mathbf{I} - \mathbf{P}_\mathbf{a}$

Basis of Projections

- Projection with arbitrary basis
  - Given any matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ that has full rank $m \geq n$

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$$

    is an orthogonal projection
  - What does $\mathbf{P}$ project onto?
    - range($\mathbf{A}$)
  - $(\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$ is called the pseudo inverse of $\mathbf{A}$, denoted as $\mathbf{A}^+$

QR Factorization

Motivation

- Question: Given a linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$ has full rank, how to solve the linear system?
- Answer: One possible solution is to use SVD. How?

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*, \text{so } \mathbf{x} = \mathbf{V}\Sigma^{-1}\mathbf{U}^*\mathbf{b}$$

Motivation

- Question: Given a linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$ has full rank, how to solve the linear system?
- Answer: One possible solution is to use SVD. How?

$$\mathbf{A} = \mathbf{U\Sigma V}^*, \text{so } \mathbf{x} = \mathbf{V\Sigma}^{-1}\mathbf{U}^*\mathbf{b}$$

Another solution is to use QR factorization, which decompose $\mathbf{A}$ into product of two simple matrices $\mathbf{Q}$ and $\mathbf{R}$ where columns of $\mathbf{Q}$ are orthonormal and $\mathbf{R}$ is upper triangular.

Two Different Versions of QR

- Full QR factorization: $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$

$$\mathbf{A} = \mathbf{QR}$$

  where $\mathbf{Q} \in \mathbb{C}^{m \times m}$ is unitary and $\mathbf{R} \in \mathbb{C}^{m \times n}$ is upper trianglular
- Reduced QR factorization: $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$

$$\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$$

  where $\hat{\mathbf{Q}} \in \mathbb{C}^{m \times n}$ contains orthonormal vectors and $\hat{\mathbf{R}} \in \mathbb{C}^{n \times n}$ is upper triangular
- What space do $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_j, j \leq n$ span?

Two Different Versions of QR

- Full QR factorization: $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$

$$\mathbf{A} = \mathbf{QR}$$

  where $\mathbf{Q} \in \mathbb{C}^{m \times m}$ is unitary and $\mathbf{R} \in \mathbb{C}^{m \times n}$ is upper trianglular

- Reduced QR factorization: $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$

$$\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$$

  where $\hat{\mathbf{Q}} \in \mathbb{C}^{m \times n}$ contains orthonormal vectors and $\hat{\mathbf{R}} \in \mathbb{C}^{n \times n}$ is upper triangular

- What space do $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_j, j \leq n$ span?
  - Answer: For full rank $\mathbf{A}$, first j column vectors of $\mathbf{A}$, i.e. $\langle \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_j \rangle = \langle \mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_j \rangle$

Gram-Schmidt Orthogonalization

- A method to construct QR factorization is to orthogonalize the column vectors of $\mathbf{A}$:
- Basic idea:
  - Take first column $\mathbf{a}_1$ and normalize it to obtain vector $\mathbf{q}_1$;
  - Take second column $\mathbf{a}_2$, subtract its orthogonal projection to $\mathbf{q}_1$, and normalize to obtain $\mathbf{q}_2$;
  - $\cdots$
  - Take j-th column of $\mathbf{a}_j$, subtract its orthogonal projection to $\mathbf{q}_1, \cdots, \mathbf{q}_{j-1}$ and normalize to obtain $\mathbf{q}_j$

$$\mathbf{v}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i^* \mathbf{a}_j \mathbf{q}_i, \quad \mathbf{q}_j = \mathbf{v}_j / \|\mathbf{v}_j\|$$

- This idea is called Gram-Schmidt orthogonalization.

Gram Schmidt Projections

- Orthogonal vectors produced by Gram-Schmidt can be written in terms of projectors

$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|}$$

where

$$\mathbf{P}_j = \mathbf{I} - \hat{\mathbf{Q}}_{j-1}\hat{\mathbf{Q}}_{j-1}^* \text{ with } \hat{\mathbf{Q}}_{j-1} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \mathbf{q}_{j-1}]$$

- $\mathbf{P}_j$ projects orthogonally onto space orthogonal to $\langle \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{j-1} \rangle$ and rank of $\mathbf{P}_j$ is $m - (j-1)$

Algorithm of Gram Schmidt Orthogonalization

Classical Gram-Schmidt method

for $j = 1$ **to** n
  $\mathbf{v}_j = \mathbf{a}_j$;
  for $i = 1$ **to** $j-1$
    $r_{ij} = \mathbf{q}_i^* \mathbf{a}_j$
    $\mathbf{v}_j = \mathbf{v}_j - r_{ij}\mathbf{q}_i$
  $r_{jj} = \|\mathbf{v}_j\|_2$
  $\mathbf{q}_j = \frac{\mathbf{v}_j}{r_{jj}}$

- Classical Gram-Schmidt (CGS) is unstable, which means that its solution is sensitive to perturbation

Existence of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$ has full QR factorization, hence also a reduced QR factorization.

Key idea of proof:

- If $\mathbf{A}$ has full rank, Gram-Schmidt algorithm provides a proof itself for having reduced QR.
- If $\mathbf{A}$ does not have full rank, at some step $\mathbf{v}_j = 0$. We can set $\mathbf{q}_j$ to be a vector orthogonal to $\mathbf{q}_i, i < j$.
- To construct full QR from reduced QR, just continue Gram-Schmidt an additional $m - n$ steps.

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$
Question: Is full QR factorization unique?

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$
Question: Is full QR factorization unique?
Question: What if A does not have full rank?

Gram-Schmidt Orthogonalization

Gram-Schmidt Orthogonalization

- A method to construct QR factorization is to orthogonalize the column vectors of $\mathbf{A}$:
- Basic idea:
  - Take first column $\mathbf{a}_1$ and normalize it to obtain vector $\mathbf{q}_1$;
  - Take second column $\mathbf{a}_2$, subtract its orthogonal projection to $\mathbf{q}_1$, and normalize to obtain $\mathbf{q}_2$;
  - $\cdots$
  - Take j-th column of $\mathbf{a}_j$, subtract its orthogonal projection to $\mathbf{q}_1, \cdots, \mathbf{q}_{j-1}$ and normalize to obtain $\mathbf{q}_j$

$$\mathbf{v}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} \mathbf{q}_i^* \mathbf{a}_j \mathbf{q}_i, \quad \mathbf{q}_j = \mathbf{v}_j / \|\mathbf{v}_j\|$$

- This idea is called Gram-Schmidt orthogonalization.

Gram Schmidt Projections

- Orthogonal vectors produced by Gram-Schmidt can be written in terms of projectors

$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|}$$

where

$$\mathbf{P}_j = \mathbf{I} - \hat{\mathbf{Q}}_{j-1}\hat{\mathbf{Q}}_{j-1}^* \text{ with } \hat{\mathbf{Q}}_{j-1} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \mathbf{q}_{j-1}]$$

- $\mathbf{P}_j$ projects orthogonally onto space orthogonal to $\langle \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{j-1} \rangle$ and rank of $\mathbf{P}_j$ is $m-(j-1)$

Algorithm of Gram Schmidt Orthogonalization

Classical Gram-Schmidt method

for $j = 1$ **to** n
  $\mathbf{v}_j = \mathbf{a}_j$;
  for $i = 1$ **to** $j-1$
    $r_{ij} = \mathbf{q}_i^* \mathbf{a}_j$
    $\mathbf{v}_j = \mathbf{v}_j - r_{ij}\mathbf{q}_i$
  $r_{jj} = \|\mathbf{v}_j\|_2$
  $\mathbf{q}_j = \frac{\mathbf{v}_j}{r_{jj}}$

- Classical Gram-Schmidt (CGS) is unstable, which means that its solution is sensitive to perturbation

Existence of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m \times n} (m \geq n)$ has full QR factorization, hence also a reduced QR factorization.

Key idea of proof:

- If $\mathbf{A}$ has full rank, Gram-Schmidt algorithm provides a proof itself for having reduced QR.
- If $\mathbf{A}$ does not have full rank, at some step $\mathbf{v}_j = 0$. We can set $\mathbf{q}_j$ to be a vector orthogonal to $\mathbf{q}_i, i < j$.
- To construct full QR from reduced QR, just continue Gram-Schmidt an additional $m - n$ steps.

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m \times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$
Question: Is full QR factorization unique?

Uniqueness of QR

Theorem

Every $\mathbf{A} \in \mathbb{C}^{m\times n}(m \geq n)$ has full rank has a unique reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ with $r_{jj} > 0$.

Proof is provided by Gram-Schmidt iteration itself. If the signs of r_{jj} are determined, then r_{ij} and $\mathbf{q}_j$ are determined.
Question: Why do we require $r_{jj} > 0$
Question: Is full QR factorization unique?
Question: What if A does not have full rank?

Alternative view to Gram-Schmidt Projection

- Orthogonal vectors produced by Gram-Schmidt can be written in terms of projectors
$$\mathbf{q}_j = \frac{\mathbf{P}_j \mathbf{a}_j}{\|\mathbf{P}_j \mathbf{a}_j\|},$$
where $\mathbf{P}_j = \mathbf{I} - \hat{\mathbf{Q}}_{j-1}\hat{\mathbf{Q}}_{j-1}^*$ with $\hat{\mathbf{Q}}_{j-1} = [\mathbf{q}_1 \ \mathbf{q}_2 \ldots \mathbf{q}_{j-1}]$
- We may view $\mathbf{P}_j$ as product of a sequence of projections
$$\mathbf{P}_j = \mathbf{P}_{\perp_{q_{j-1}}} \mathbf{P}_{\perp_{q_{j-2}}} \ldots \mathbf{P}_{\perp_{q_1}}$$
where $\mathbf{P}_{\perp_q} = \mathbf{I} - \mathbf{q}\mathbf{q}^*$
- Instead of computing $\mathbf{v}_j = \mathbf{P}_j \mathbf{a}_i$, one could compute $\mathbf{v}_j = \mathbf{P}_{\perp_{q_{j-1}}} \mathbf{P}_{\perp_{q_{j-2}}} \ldots \mathbf{P}_{\perp_{q_1}} a_j$ instead, resulting in modified Gram-Schmidt algorithm

Modified Gram-Schmidt Orthogonalization

Classical Gram-Schmidt method:

for $j = 1$ **to** n

$\quad \mathbf{v}_j = \mathbf{a}_j$;

$\quad$ **for** $i = 1$ **to** $j - 1$

$\quad\quad r_{ij} = \mathbf{q}_i^* \mathbf{a}_j$

$\quad\quad \mathbf{v}_j = \mathbf{v}_j - r_{ij} \mathbf{q}_i$

$\quad r_{jj} = \|\mathbf{v}_j\|_2$

$\quad \mathbf{q}_j = \frac{\mathbf{v}_j}{r_{jj}}$

Modified Gram-Schmidt method:

for $j = 1$ to n

$\quad \mathbf{v}_j = \mathbf{a}_j$

for $i = 1$ to n

$\quad r_{ii} = \|\mathbf{v}_i\|_2$

$\quad \mathbf{q}_i = \mathbf{v}_i / r_{ii}$

$\quad$ for $j = i + 1$ to n

$\quad\quad r_{ij} = \mathbf{q}_i^* \mathbf{v}_j$

$\quad\quad \mathbf{v}_j = \mathbf{v}_j - r_{ij} \mathbf{q}_i$

Modified Gram-Schmidt Orthogonalization

Classical Gram-Schmidt method:
for $j = 1$ **to** n
  $\mathbf{v}_j = \mathbf{a}_j$;
  for $i = 1$ **to** $j-1$
    $r_{ij} = \mathbf{q}_i^* \mathbf{a}_j$
    $\mathbf{v}_j = \mathbf{v}_j - r_{ij}\mathbf{q}_i$
  $r_{jj} = \|\mathbf{v}_j\|_2$
  $\mathbf{q}_j = \frac{\mathbf{v}_j}{r_{jj}}$

Modified Gram-Schmidt method:
for $j = 1$ to n
  $\mathbf{v}_j = \mathbf{a}_j$
for $i = 1$ to n
  $r_{ii} = \|\mathbf{v}_i\|_2$
  $\mathbf{q}_i = \mathbf{v}_i / r_{ii}$
  for $j = i+1$ to n
    $r_{ij} = \mathbf{q}_i^* \mathbf{v}_j$
    $\mathbf{v}_j = \mathbf{v}_j - r_{ij}\mathbf{q}_i$

- Key difference between CGS and MGS is how r_{ij} is computed
- CGS above is column-oriented (in the sense that R is computed column by column) and MGS above is row-oriented, but this is NOT the main difference between CGS and MGS. There are also column-oriented MGS and row-oriented CGS.
- MGS is numerically more stable than CGS (less sensitive to round-off errors)

Example: CGS vs. MGS

- Consider matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}$$

where ε is small such that $1+\varepsilon^2 = 1$ with round-off error

- For both CGS and MGS

$$\begin{aligned}
\mathbf{v}_1 \leftarrow (1,\varepsilon,0,0)^T, r_{11} = \sqrt{1+\varepsilon^2} \approx 1, \mathbf{q}_1 = \mathbf{v}_1/r_{11} = (1,\varepsilon,0,0)^T, \\
\mathbf{v}_2 \leftarrow (1,0,\varepsilon,0)^T, r_{12} = \mathbf{q}_1^T\mathbf{a}_2 (or = \mathbf{q}_1^T\mathbf{v}_2) = 1 \\
\mathbf{v}_2 \leftarrow \mathbf{v}_2 - r_{12}\mathbf{q}_1 = (0,-\varepsilon,\varepsilon,0)^T \\
r_{22} = \sqrt{2}\varepsilon, \mathbf{q}_2 = (0,-1,1,0)/\sqrt{2}, \\
\mathbf{v}_3 \leftarrow (1,0,0,\varepsilon)^T, r_{13} = \mathbf{q}_1^T\mathbf{a}_3 (or = \mathbf{q}_1^T\mathbf{v}_3) = 1 \\
\mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{13}\mathbf{q}_1 = (0,-\varepsilon,0,\varepsilon)^T
\end{aligned}$$

Example: CGS vs. MGS Cont'd

- For CGS:

$$r_{23} = \mathbf{q}_2^T \mathbf{a}_3 = 0, \mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{23}\mathbf{q}_2 = (0, -\varepsilon, 0, \varepsilon)^T$$
$$r_{33} = \sqrt{2}\varepsilon, \mathbf{q}_3 = \mathbf{v}_3 / r_{33} = (0, -1, 0, 1)^T / \sqrt{2}$$

  - Note that $\mathbf{q}_2^T \mathbf{q}_3 = (0, -1, 1, 0))0, -1, 0, 1)^T / 2 = 1/2$

- For MGS:

$$r_{23} = \mathbf{q}_2^T \mathbf{a}_3 = \varepsilon/\sqrt{2}, \mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{23}\mathbf{q}_2 = (0, -\varepsilon/2, -\varepsilon/2, \varepsilon)^T$$
$$r_{33} = \sqrt{6}\varepsilon/2, \mathbf{q}_3 = \mathbf{v}_3 / r_{33} = (0, -1, -1, 2)^T / \sqrt{6}$$

  - Note that $\mathbf{q}_2^T \mathbf{q}_3 = (0, -1, 1, 0)(0, -1, -1, 2)^T / \sqrt{12} = 0$

Operation Count

- It is important to assess the efficiency of algorithms. But how?
  - We could implement different algorithms and do head-to-head comparison, but implementation details might affect true performance
  - We could estimate cost of all operations, but it is very tedious
  - Relatively simple and effective approach is to estimate amount of floating-point operations, or 'flops', and focus on asymptotic analysis as sizes of matrices approach infinity
- Count each operation $+, -, *, /$, and $\sqrt{}$ as one flop, and make no distinction of real and complex numbers

Theorem

CGS and MGS require $\sim 2mn^2$ flops to compute a QR factorization of an $m \times n$ matrix.