1. Finding $Q_j$ such that $\|Q_j Q_j^* A - A\| < \epsilon$ is guaranteed

A shortcoming of the last algorithm discussed in the previous lecture is that the resulting matrix $Q$ is not guaranteed to satisfy $\|Q Q^* A - A\| < \epsilon$; it merely achieves this requirement with a high degree of probability. In the case that we require certainty in our result, another approach is required. One such approach is given by the following algorithm:

- Algorithm 1
  - Given inputs $A$ and $\epsilon > 0$, set $A_0 = A$, $B_0 = [\ ]$, $Q_0 = [\ ]$.
  - for $j = 1, 2, \ldots$
    - Choose an i.i.d. $w \in \mathbb{R}^n$, set $q_j = Aw/\|Aw\|$.
    - Set $b_j = q_j^* A_{j-1}$.
    - Set $Q_j = [Q_{j-1} q_j]$.
    - Set $B_j = \begin{bmatrix} B_{j-1} \\ b_j \end{bmatrix}$.
    - Set $A_j = A_{j-1} - q_j b_j = A_{j-1} - q_j q_j^* A_{j-1}$.
    - If $\|A_j\| < \epsilon$, exit.

We will next show that the algorithm “works,” that is, we will show that $A_j = Q_j Q_j^* A - A$, so when we terminate the routine (i.e. when $\|A_j\| < \epsilon$), we also have $\|Q_j Q_j^* A - A\| < \epsilon$. We will do this by induction.

**Iteration 1.** Here we have $Q_1 = [q_1]$ and $A_0 = A$, so

$$A_1 = A_0 - q_1 q_1^* A_0 = (I - Q_1 Q_1^*) A.$$ 

**Iteration $j$/Induction Step.** Assume we have $A_j = (I - Q_j Q_j^*) A$.

$$A_{j+1} = A_j - q_{j+1} q_j^* A_j = (A - Q_j Q_j^* A) - q_{j+1} q_{j+1}^* (A - Q_j Q_j^* A).$$

Now since $q_{j+1}$ is orthogonal to the columns of $Q_j$, $q_j^* Q_j = 0$. Therefore

$$(A - Q_j Q_j^* A) - q_{j+1} q_{j+1}^* (A - Q_j Q_j^* A) = A - Q_j Q_j^* A - q_{j+1} q_{j+1}^* A$$

$$= (I - (Q_j Q_j^* + q_{j+1} q_{j+1}^*)) A.$$ 

Then since

$$Q_{j+1} Q_{j+1}^* = \begin{bmatrix} Q_j \\ q_{j+1} \end{bmatrix} \begin{bmatrix} Q_j^* & q_{j+1}^* \end{bmatrix} = Q_j Q_j^* + q_{j+1} q_{j+1}^*,$$

we have that

$$A_{j+1} = (I - Q_{j+1} Q_{j+1}^*) A.$$
2. Blocking Algorithm 1

One disadvantage of Algorithm 1 is that the matrix $Q$ is updated one vector at a time, so the algorithm relies mostly on matrix-vector products rather than the more efficient matrix-matrix products. The following modification of Algorithm 1 remedies this problem.

- **Algorithm 2**
  - Set $A_0 = A$, $B_0 = [ ]$, $Q_0 = [ ]$.
  - for $j = 1, 2, \ldots$
    * Set $Q_j = \text{qr}(A_{j-1} \Omega_{j-1}, 0)$, where $\Omega \in \mathbb{R}^{n \times b}$ matrix whose entries are i.i.d. Gaussian random variables, where $b$ is some block size.
    * Set $B_j = \hat{Q}_j A_{j-1}$.
    * Set $A_j = A_{j-1} - Q_j B_j$.
    * Set $B_j = [B_j^* \; \hat{B}_j^*]$, $Q_j = [Q_{j-1} \; \hat{Q}_j]$.
    * If $\|A_j\| < \epsilon$, exit.

We would again like to show that the algorithm gives the expected result, which means we must show that $A_j = (I - Q_j Q_j^*) A$.

We will first demonstrate that $\hat{Q}_j^* \hat{Q}_k = 0$ for $k = 1, \ldots, j - 1$. The desired result will then follow. We have that $\text{Ran}(\hat{Q}_j) = \text{Ran}(A_{j-1} \Omega) \subseteq \text{Ran}(A_{j-1}) = \text{Ran}(I - \sum_{k=1}^{j-1} \hat{Q}_k \hat{Q}_k^*) \subseteq \text{Ran}(I - \sum_{k=1}^{j-1} Q_k Q_k^*) = \text{Ran}(\sum_{k=1}^{j-1} \hat{Q}_k \hat{Q}_k^*)^\perp$. Thus $\hat{Q}_j^* Q_k = 0$ for $k = 1, \ldots, j - 1$.

Now we have that

$$ A_j = (I - \hat{Q}_j \hat{Q}_j^*) A_{j-1} $$

$$ = (I - \hat{Q}_j \hat{Q}_j^*)(I - \hat{Q}_{j-1} \hat{Q}_{j-1}^*) A_{j-2} $$

$$ = \ldots = (I - \hat{Q}_j \hat{Q}_j^* - \hat{Q}_{j-1} \hat{Q}_{j-1}^* - \ldots - \hat{Q}_1 \hat{Q}_1^*) A $$

$$ = (I - Q_j Q_j^*) A. $$