# Homework 9

## This homework is due on Friday, April 1, 2022, at 11:59PM. Self-grades and HW Resubmissions are due on the following Friday, April 8, 2022, at 11:59PM.

1. **Reading Lecture Notes**

   Staying up to date with lectures is an important part of the learning process in this course. Here are links to the notes that you need to read for this week: Note 13 Note 15.

   (a) Consider $A \in \mathbb{R}^{n\times n}$ where the columns of $A$ (denoted by $\vec{a}_k, 1 \leq k \leq n$) are orthonormal. **What does the least squares solution** $(A^\top A)^{-1}A^\top \vec{y}$ **simplify to?**

   (b) Suppose we have two vectors $\vec{v}_1 = \begin{bmatrix}1\\1\end{bmatrix}$ and $\vec{v}_2 = \begin{bmatrix}1\\2\end{bmatrix} \in \mathbb{R}^2$. Then, we use Gram-Schmidt orthonormalization to construct $\vec{q}_1$ and $\vec{q}_2$. **Are** $\vec{q}_1$ **and** $\vec{q}_2$ **the only vectors that form an orthogonal basis for** $\mathrm{Span}(\vec{v}_1, \vec{v}_2)$**?**

   (c) Give a brief outline for how you would compute the Schur decomposition (i.e. upper-triangularization) of some general square matrix under the assumption that all eigenvalues are real.

   (d) What happens when we upper-triangularize a symmetric matrix?

**2. Gram-Schmidt Basics**

(a) **Use Gram-Schmidt to find a matrix $U$ whose columns form an orthonormal basis for the column space of $V$.**

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

(b) Show that you get the same resulting vector when you project $\vec{w} = \begin{bmatrix} 1 \\ -1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$ onto the columns of $V$ as you do when you project onto the columns of $U$, i.e. **show that**

$$V(V^\top V)^{-1}V^\top \vec{w} = U(U^\top U)^{-1}U^\top \vec{w}. \tag{2}$$

Feel free to use `NumPy` for the projection onto the columns of $V$, but compute the projection onto the columns of $U$ by hand. Comment on whether projecting upon the $V$ or $U$ basis is computationally more efficient. *(HINT: Which of these matrices allow us to circumvent the matrix inversion in the projection formula?)*

**3. Upper Triangularization**

In this problem, you need to upper-triangularize the matrix

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \tag{3}$$

The eigenvalues of this matrix $A$ are $\lambda_1 = \lambda_2 = 2$ and $\lambda_3 = -4$. We want to express $A$ as

$$A = \begin{bmatrix} \vec{x} & \vec{y} & \vec{z} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & a & b \\ 0 & \lambda_2 & c \\ 0 & 0 & \lambda_3 \end{bmatrix} \cdot \begin{bmatrix} \vec{x}^\top \\ \vec{y}^\top \\ \vec{z}^\top \end{bmatrix} \tag{4}$$

where $\vec{x}, \vec{y}, \vec{z}$ are orthonormal. Your goal in this problem is to compute $\vec{x}, \vec{y}, \vec{z}$ so that they satisfy the above relationship for some constants $a, b, c$.

Here are some potentially useful facts that we have gathered to save you some computations, you'll have to grind out the rest yourself.

$$\begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ \sqrt{2} \\ 0 \end{bmatrix}. \tag{5}$$

We also know that

$$\left\{ \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \tag{6}$$

is an orthonormal basis, and

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -2\sqrt{2} \\ -2\sqrt{2} & -2 \end{bmatrix}. \tag{7}$$

We also know that $\begin{bmatrix} 0 & -2\sqrt{2} \\ -2\sqrt{2} & -2 \end{bmatrix}$ has eigenvalues 2 and $-4$. The normalized eigenvector corresponding to $\lambda = 2$ is $\begin{bmatrix} -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}$. A vector which is orthogonal to that normalized eigenvector and is itself normalized is $\begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{6}}{3} \end{bmatrix}$.

**Based on the above information, compute $\vec{x}, \vec{y}, \vec{z}$.** Show your work.

You don't have to compute the constants $a, b, c$ in the interests of time.

**4. Using Upper-Triangularization to Solve Differential Equations**

You know that for any square matrix $A$ with real eigenvalues, there exists a real matrix $V$ with orthonormal columns and a real upper triangular matrix $R$ so that $A = VRV^{\top}$. In particular, to set notation explicitly:

$$V = \left[\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n\right] \tag{8}$$

$$R = \begin{bmatrix} \vec{r}_1^{\top} \\ \vec{r}_2^{\top} \\ \vdots \\ \vec{r}_n^{\top} \end{bmatrix} \tag{9}$$

where the rows of the upper-triangular $R$ look like

$$\vec{r}_1^{\top} = \begin{bmatrix}\lambda_1 & r_{1,2} & r_{1,3} & \dots & r_{1,n}\end{bmatrix} \tag{10}$$

$$\vec{r}_2^{\top} = \left[0, \lambda_2, r_{2,3}, r_{2,4}, \dots \quad r_{2,n}\right] \tag{11}$$

$$\vec{r}_i^{\top} = \left[\underbrace{0, \dots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \dots, r_{i,n}\right] \tag{12}$$

$$\vec{r}_n^{\top} = \left[\underbrace{0, \dots, 0}_{n-1 \text{ times}}, \lambda_n\right] \tag{13}$$

where the $\lambda_i$ are the eigenvalues of $A$.

Suppose our goal is to solve the $n$-dimensional system of differential equations written out in vector/matrix form as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{14}$$

$$\vec{x}(0) = \vec{x}_0, \tag{15}$$

where $\vec{x}_0$ is a specified initial condition and $\vec{u}(t)$ is a given vector of functions of time.

Assume that the $V$ and $R$ have already been computed and are accessible to you using the notation above.

Assume that you have access to a function $\texttt{ScalarSolve}(\lambda, y_0, \check{u})$ that takes a real number $\lambda$, a real number $y_0$, and a real-valued function of time $\check{u}$ as inputs and returns a real-valued function of time that is the solution to the scalar differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \lambda y(t) + \check{u}(t) \tag{16}$$

with initial condition $y(0) = y_0$.

Also assume that you can do regular arithmetic using real-valued functions and it will do the right thing. So if $u$ is a real-valued function of time, and $g$ is also a real-valued function of time, then $5u + 6g$ will be a real valued function of time that evaluates to $5u(t) + 6g(t)$ at time $t$.

**Use $V, R$ to construct a procedure for solving this differential equation**

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{17}$$

$$\vec{x}(0) = \vec{x}_0, \tag{18}$$

**for $\vec{x}(t)$ by filling in the following template in the spots marked $\clubsuit, \diamondsuit, \heartsuit, \spadesuit$.**

*NOTE*: It will be useful to upper triangularize $A$ by change of basis to get a differential equation in terms of $R$ instead of $A$.

*(HINT: The process here should be similar to diagonalization with some modifications. Start from the last row of the system and work your way up to understand the algorithm.)*

1: $\vec{\widetilde{x}}_0 = V^\top \vec{x}_0$ ▷ Change the initial condition to be in $V$-coordinates
2: $\vec{\widetilde{u}} = V^\top \vec{u}$ ▷ Change the external input functions to be in $V$-coordinates
3: **for** $i = n$ down to 1 **do** ▷ Iterate up from the bottom row
4: $\quad \check{u}_i = \clubsuit + \sum_{j=i+1}^{n} \spadesuit$ ▷ Make the effective input for this level
5: $\quad \widetilde{x}_i = \texttt{ScalarSolve}(\diamondsuit, \widetilde{x}_{0,i}, \check{u}_i)$ ▷ Solve this level's scalar differential equation
6: **end for**
7: $\vec{x}(t) = \heartsuit \begin{bmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \\ \vdots \\ \widetilde{x}_n \end{bmatrix}(t)$ ▷ Change back into original coordinates

(a) **Give the expression for $\heartsuit$ on line 7 of the algorithm above.** (i.e., how do you get from $\vec{\widetilde{x}}(t)$ to $\vec{x}(t)$?)

(b) **Give the expression for $\diamondsuit$ on line 5 of the algorithm above.** (i.e., what are the $\lambda$ arguments to `ScalarSolve`, equation (2), for the $i^{\text{th}}$ iteration of the for-loop?)

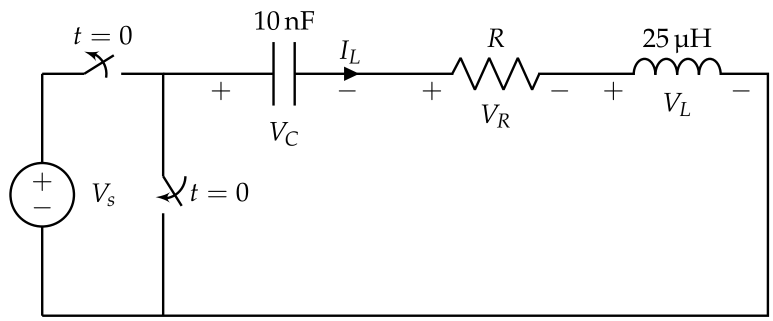
*(HINT: Convert the differential equation to be in terms of $R$ instead of $A$. It may be helpful to start with $i = n$ and develop a general form for the $i^{\text{th}}$ row.)*

(c) **Give the expression for $\clubsuit$ on line 4 of the algorithm above.**

(d) **Give the expression for $\spadesuit$ on line 4 of the algorithm above.**

**5. RLC Responses: Critically Damped Case**

Recall the series RLC circuit we worked on in Homework 4. We're going to re-visit this problem for a special case: the critically-damped case. (Notice $R$ is not specified yet. You'll have to figure out what that is.)




Assume the circuit above has reached steady state for $t < 0$. At time $t = 0$, the switch changes state and disconnects the voltage source, replacing it with a short.

For this problem, you may use a calculator that can handle matrices (Matlab, Mathematica, Wolfram Alpha) or the attached `RLC_Calc.ipynb` Jupyter Notebook for numerical calculations.

Recall from Homework 4 that we were able to represent this series RLC circuit as a system of equations in vector/matrix form. We let the vector state variable $\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix}$ and were able to write the system in the form $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t)$ with a $2 \times 2$ matrix $A$:

$$\begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t}x_1(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}x_2(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \tag{19}$$

where

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}. \tag{20}$$

Further, we showed that the two eigenvalues of $A$ are:

$$\lambda_1 = -\frac{1}{2}\frac{R}{L} + \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{21}$$

$$\lambda_2 = -\frac{1}{2}\frac{R}{L} - \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{22}$$

(a) **For what value of $R$ are the two eigenvalues of $A$ going to be identical?** We will refer to this value as $R_{\text{crit}}$ later on.

(b) Using the given values for the capacitor and the inductor, as well as $R_{\text{crit}}$ you found in the previous part, **find the eigenvalues and eigenspaces of $A$. What are the dimensions of the corresponding eigenspaces?** (i.e. how many linearly independent eigenvectors can you find associated with this eigenvalue?)

It may be easier to plug in numbers into the matrix first.

You should have found above that the dimension of the eigenspace is 1, i.e. the eigenvectors are linearly dependent. Herein lies the problem. In the overdamped, underdamped, and undamped RLC cases we saw previously in Homework 4, we were able to solve for a system of equation by conducting a coordinate transformation using the eigenvectors of $A$. In other words, we were able to construct a matrix $V = \begin{bmatrix} \vec{v}_{\lambda_1} & \vec{v}_{\lambda_2} \end{bmatrix}$ where $\vec{v}_{\lambda_1}$ and $\vec{v}_{\lambda_2}$ are the linearly independent eigenvalues of $A$. We could then diagonalize the matrix $A$ as $\widetilde{A} = V^{-1}AV = \Lambda$ and solve for the transformed system. The problem we have now is that the eigenvectors $\vec{v}_{\lambda_1}$ and $\vec{v}_{\lambda_2}$ of $A$ are not linearly dependent, so $V = \begin{bmatrix} \vec{v}_{\lambda_1} & \vec{v}_{\lambda_2} \end{bmatrix}$ is not invertible. Therefore, $A$ is not diagonalizable.

The downside here is we can't diagonalize $A$ to convert our matrix-vector equation to a system of uncoupled scalar equations, which we could solve for independently for each of our state variables. However, we know that even if we can't diagonalize $A$, we can still upper triangularize it. This yields a system of chained scalar equations, which we can use to solve for our state variables via back-substitution.

To triangularize, we first must find an appropriate change of basis matrix $U = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 \end{bmatrix}$. This lets us calculate our triangular matrix $T = U^{-1}AU$. If $U$ is an orthonormal matrix (orthogonal column vectors which are normalized), then we can exploit the fact that $U^{-1} = U^T$ and calculate $T = U^T AU$.

(c) There are multiple ways to find an upper triangular matrix of $A$, and it is not unique. Regardless of your previous result, assume the system matrix is:

$$A = \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \tag{23}$$

If you use the Real Schur Decomposition algorithm from Note 15, you would find an upper triangular matrix $T$ and the associated basis $U$ for the system matrix $A$ given above. For brevity, we will provide you with the basis $U$:

$$U = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{24}$$

Note that $U$ is an orthonormal matrix. **Find the associated triangular matrix $T$.** You may use your favorite matrix calculator, e.g. Python, Jupyter notebook, MATLAB, Mathematica, Wolfram Alpha, etc.

(d) We have solved for a coordinate system $U$ which triangularizes our system matrix $A$ to the $T$ we found. We apply a change of basis to define $\vec{\widetilde{x}}$ in the transformed coordinates such that $\vec{x}(t) = U\vec{\widetilde{x}}(t)$. **Starting from $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t)$, show that the differential equation in the transformed coordinate system is $\frac{\mathrm{d}}{\mathrm{d}t}\vec{\widetilde{x}}(t) = T\vec{\widetilde{x}}(t)$**

(e) Notice that the second differential equation for $\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{x}_2(t)$ in the above coordinate system only depends on $\widetilde{x}_2(t)$ itself. There is no cross-term dependence. This happened because we transformed into a coordinate system which triangularizes $A$. **Compute the initial condition for $\widetilde{x}_2(0)$ and write out the solution to this scalar differential equation for $\widetilde{x}_2(t)$ for $t \geq 0$.** Assume that $V_s = 1\,\mathrm{V}$. Recall that, in our original system, $\vec{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix}$.

(f) With an explicit solution to $\widetilde{x}_2(t)$ in hand, use this to back-substitute and **write out the resulting scalar differential equation for $\widetilde{x}_1(t)$.** This should effectively have a time-dependent input (i.e. $\widetilde{x}_2(t)$) in it. **Also compute the initial condition for $\widetilde{x}_1(0)$** (which you may have already solved for in the previous part).

(g) The differential equation you found should look like a standard form seen in a previous homework with a non-zero input. The input here happens to be exponential. **Solve the above scalar differential equation for $\widetilde{x}_1(t)$ over $t \geq 0$.** *(HINT: See Homework 2, Question 4.)*

(h) **Find $x_1(t)$ and $x_2(t)$ for $t \geq 0$ based on the answers to the previous three parts.**

(i) In the `RLCSliders.ipynb` Jupyter notebook, move the resistance slider to your $R_{\text{crit}}$ value. Next, move the slider so that the resistance is slightly lower — yielding an underdamped system — and slightly higher – yielding an overdamped system (both of which you already solved for in Homework 4). **Qualitatively describe the settling response and the eigenvalues for the underdamped, critically damped, and overdamped cases. Is there any ringing in the time response? Are the eigenvalues real, complex conjugates, identical?** For the underdamped and overdamped cases, you can use any resistance value. For the critically damped case, pick your resistance precisely.

**6. (OPTIONAL) Correctness of the Gram-Schmidt Algorithm**

*NOTE*: This problem is optional because we wanted to make the homework shorter. However, all the concepts covered are in scope, and are fair game for future content, including the final exam.

Suppose we take a list of vectors $\{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n\}$ and run the following Gram-Schmidt algorithm on it to perform orthonormalization. It produces the vectors $\{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$.

**for** $i = 1$ up to $n$ **do** ▷ Iterate through the vectors
    $\vec{r}_i = \vec{a}_i - \sum_{j<i} \vec{q}_j \left(\vec{q}_j^\top \vec{a}_i\right)$ ▷ Find the amount of $\vec{a}_i$ that remains after we project
    **if** $\vec{r}_i = \vec{0}$ **then**
        $\vec{q}_i = \vec{0}$
    **else**
        $\vec{q}_i = \frac{\vec{r}_i}{\|\vec{r}_i\|}$ ▷ Normalize the vector.
    **end if**
**end for**

In this problem, we prove the correctness of the Gram-Schmidt algorithm by showing that the following three properties hold on the vectors output by the algorithm.

1. If $\vec{q}_i \neq \vec{0}$, then $\vec{q}_i^\top \vec{q}_i = \|\vec{q}_i\|^2 = 1$ (i.e. the $\vec{q}_i$ have unit norm whenever they are nonzero).
2. For all $1 \leq \ell \leq n$, $\mathrm{Span}(\{\vec{a}_1, \ldots, \vec{a}_\ell\}) = \mathrm{Span}(\{\vec{q}_1, \ldots, \vec{q}_\ell\})$.
3. For all $i \neq j$, $\vec{q}_i^\top \vec{q}_j = 0$ (i.e. $\vec{q}_i$ and $\vec{q}_j$ are orthogonal).

(a) First, we show that the first property holds by construction from the if/then/else statement in the algorithm. It holds when $\vec{q}_i = \vec{0}$, since the first property has no restrictions on $\vec{q}_i$ if it is the zero vector. **Show that** $\|\vec{q}_i\| = 1$ **if** $\vec{q}_i \neq \vec{0}$.

(b) Next, we show the second property by considering each $\ell$ from 1 to $n$, and showing the statement that $\mathrm{Span}(\{\vec{a}_1, \ldots, \vec{a}_\ell\}) = \mathrm{Span}(\{\vec{q}_1, \ldots, \vec{q}_\ell\})$. This statement is true when $\ell = 1$ since the algorithm produces $\vec{q}_1$ as a scaled version of $\vec{a}_1$. Now assume that this statement is true for $\ell = k-1$. Under this assumption, **show that the spans are the same for** $\ell = k$.

This implies that because $\mathrm{Span}(\{\vec{a}_1\}) = \mathrm{Span}(\{\vec{q}_1\})$, then so too is $\mathrm{Span}(\{\vec{a}_1, \vec{a}_2\}) = \mathrm{Span}(\{\vec{q}_1, \vec{q}_2\})$, and so forth, until we get that $\mathrm{Span}(\{\vec{a}_1, \ldots, \vec{a}_n\}) = \mathrm{Span}(\{\vec{q}_1, \ldots, \vec{q}_n\})$.

*(HINT: What you need to show is: if there exists* $\vec{\alpha} = \begin{bmatrix}\alpha_1 & \cdots & \alpha_k\end{bmatrix} \neq \vec{0}_k$ *so that* $\vec{y} = \sum_{j=1}^{k} \alpha_j \vec{a}_j$*, then there exists* $\vec{\beta} = \begin{bmatrix}\beta_1 & \cdots & \beta_k\end{bmatrix} \neq \vec{0}_k$ *such that* $\vec{y} = \sum_{j=1}^{k-1} \beta_j \vec{q}_j$ *(this is the forward direction). And vice versa from* $\vec{\beta}$ *to* $\vec{\alpha}$ *(this is the reverse direction).)*

*(HINT: To show the forward direction, write* $\vec{a}_k$ *in terms of* $\vec{q}_k$ *and earlier* $\vec{q}_j$*. Use the condition for* $\ell = k-1$ *to show the condition for* $\ell = k$*. Don't forget the case that* $\vec{q}_k = \vec{0}$*. The reverse direction may be approached similarly.)*

(c) Lastly, we establish orthogonality between every pair of vectors in $\{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$. Consider each $\ell$ from 1 to $n$. We want to show the statement that for all $j < \ell$, $\vec{q}_j^\top \vec{q}_\ell = 0$. The statement holds for $\ell = 1$ since there are no $j < 1$. Assume that this statement holds for $\ell$ up to and including $k-1$. That is, we assume that for all $i \leq k-1$, $\vec{q}_j^\top \vec{q}_i = 0$ for all $j < i$.

Under this assumption, **show that for all** $i \leq k$**, that** $\vec{q}_j^\top \vec{q}_i = 0$ **for all** $j < i$**.** This shows that every pair of distinct vectors up to $1, 2, \ldots, \ell$ are orthogonal for each $\ell$ from 1 to $n$.

*(HINT: The cases* $i \leq k-1$ *are already covered by the assumption. So you can focus on* $i = k$*. Next, notice that the case* $\vec{q}_k = \vec{0}$ *is also true, since the inner product of any vector with* $\vec{q}_k = \vec{0}$ *is* $\vec{0}$*. So, focus on the case* $\vec{q}_k \neq \vec{0}$ *and expand what you know about* $\vec{q}_k$*.)*

(d) **(OPTIONAL)** The second property can be summarized in matrix form. Let $A = \begin{bmatrix}\vec{a}_1 & \cdots & \vec{a}_n\end{bmatrix}$ and $Q = \begin{bmatrix}\vec{q}_1 & \cdots & \vec{q}_n\end{bmatrix}$. If $A$ and $Q$ have the same column span then there exists square matrix $U = \begin{bmatrix}\vec{u}_1 & \cdots & \vec{u}_n\end{bmatrix}$ such that $A = QU$.

**Show that a $U$ can be found that is upper-triangular (i.e. the $i$-th column $\vec{u}_i$ of $U$ has zero entries in it for every row after the $i$-th position.)**
*(HINT: Matrix multiplication tells you that $\vec{a}_i = \sum_{j=1}^{n} \vec{u}_{i_j} \vec{q}_j$. What does the algorithm tell you about this relationship? Can you figure out what $\vec{u}_{i_j}$ should be?)*

7. **(OPTIONAL) Make Your Own Problem.**

   **Write your own problem about content covered in the course thus far, and provide a thorough solution to it.**

   *NOTE*: This can be a totally new problem, a modification on an existing problem, or a Jupyter part for a problem that previously didn't have one. Please cite all sources for anything (including course material) that you used as inspiration.

   *NOTE*: High-quality problems may be used as inspiration for the problems we choose to put on future homeworks or exams.

8. **Homework Process and Study Group**

   Citing sources and collaborators are an important part of life, including being a student!

   We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

   (a) **What sources (if any) did you use as you worked through the homework?**

   (b) **If you worked with someone on this homework, who did you work with?**
   List names and student ID's. (In case of homework party, you can also just describe the group.)

   (c) **Roughly how many total hours did you work on this homework? Write it down here where you'll need to remember it for the self-grade form.**

**Contributors:**

- Kuan-Yun Lee.
- Kunmo Kim.
- Jichan Chung.
- Anant Sahai.
- Sidney Buchbinder.
- Daniel Abraham.
- Jaijeet Roychowdhury.
- Sanjeet Batra.
- Aditya Arun.
- Alex Devonport.
- Regina Eckert.
- Ashwin Vangipuram.
- Mike Danielczuk.
- Ayan Biswas.
- Wahid Rahman.
- Nathan Lambert.
- Sally Hui.