

**This homework is due on Sunday, July 24 at 11:59 pm PT.**

**1. Midterm Redo**

Congratulations on finishing the midterm!

The midterm redo is an optional assignment. However, if you want to qualify for the clobber policy, it is mandatory. As a reminder, the clobber policy grants the following:

- $\text{Post Clobber Midterm \%} = \max(\text{Final Exam \%} - 5 \%, \text{Original Midterm \%})$

What does it mean to "complete" the midterm redo assignment? We'd like to see you make an honest attempt on every problem. You will most definitely qualify for the clobber then.

There will be a designated assignment on Gradescope for the Midterm Redo due Monday, July 25, 11:59 PM PDT.

**To get credit for this problem, let us know if you will do the midterm redo or not.**

## 2. Stability for Information Processing: Solving Least-Squares via Gradient Descent with a Constant Step Size

In this problem, we will derive a dynamical system approach for solving a least-squares problem which finds the  $\vec{x}$  that minimizes  $\|A\vec{x} - \vec{y}\|^2$ . Here, we assume  $A \in \mathbb{R}^{m \times n}$  to be tall and full rank — i.e., it has linearly independent columns.

As covered in EECS16A, this has a closed-form solution:

$$\vec{x} = (A^\top A)^{-1} A^\top \vec{y}. \quad (1)$$

In many cases, direct computation of this solution is too slow, because it requires computing  $(A^\top A)^{-1}$ , which is generally very costly. We will instead solve the problem iteratively using so-called "gradient descent" which turns this particular problem into an analysis of a particular discrete-time state-space dynamical system. *Gradient descent is not in scope for 16B, but is used throughout machine learning as a means of iterately converging towards minimums by following the direction of gradients. You can learn more about it through upper-division courses like EECS 127 and CS189, but for 16B, you may assume Gradient Descent is just a special form of a discrete time function.*

- (a) **Show that  $\vec{y} - A\vec{x}$  is orthogonal to the columns of  $A$ , i.e., show  $A^\top(\vec{y} - A\vec{x}) = \vec{0}$ .**

NOTE: This was shown to you in 16A, but it is important that you see this for yourself again.

- (b) In iterative optimization schemes such as gradient descent, we will get a sequence of estimates for  $\vec{x}$  at each timestep. Let  $\vec{x}[i]$  denote our estimate for  $\vec{x}$  at timestep  $i$ .

When implementing the optimization algorithm, we choose the update rule and thus the state space equation. We would like to develop a state space equation for which the state  $\vec{x}[i]$  will converge to  $\vec{x}$ , the true least squares solution.

Gradient descent gives us the following update rule:

$$\vec{x}[i+1] = \vec{x}[i] + \alpha A^\top (\vec{y} - A\vec{x}[i]) \quad (2)$$

that gives us an updated estimate using the previous one. Here  $\alpha$  is the "step size" or "learning rate" that we choose. For us in 16B, it doesn't matter where we got the update rule, but the important thing to note is that if  $\vec{x}[i] = \vec{x}$ , then by the previous part,  $\vec{x}[i+1] = \vec{x}$  and the system remains in equilibrium at  $\vec{x}$  for all time.

To show that  $\vec{x}[i] \rightarrow \vec{x}$ , we define a new state variable  $\Delta\vec{x}[i] = \vec{x}[i] - \vec{x}$ . It represents the deviation from where we want to be.

**Derive the discrete-time state evolution equation for  $\Delta\vec{x}[i]$ , and show that it takes the form:**

$$\Delta\vec{x}[i+1] = (I - \alpha G)\Delta\vec{x}[i]. \quad (3)$$

**What is  $G$ ?**

- (c) We would like to change the learning rate  $\alpha$  such that  $\Delta\vec{x}[i]$  converges to 0. In particular, we want to make sure that we have a stable system. To do this, we need to understand the eigenvalues of  $I - \alpha G$ . **Show that the eigenvalues of matrix  $I - \alpha G$  are  $1 - \alpha\lambda_k\{G\}$ , where  $\lambda_k\{G\}$  are the eigenvalues of  $G$ , for  $k \in \{1, 2, \dots, n\}$ .**

- (d) For system eq. (3) to be stable, we need all the eigenvalues of  $I - \alpha G$  to have magnitudes that are smaller than 1 (since this is a discrete-time system). **Above what value of  $\alpha$  would the system (3) become unstable?** This is what happens if you try to set the  $\alpha$  to be too high.

*NOTE:* Since the matrix  $G$  above has a special form, all of the eigenvalues of  $G$  are non-negative and real. (You'll see why later in this course.)

- (e) **Play with the given Jupyter notebook and comment on what you observe.** How does the size of the learning rate  $\alpha$  affect the estimate? At what speed are the estimates converging?

### 3. Gram-Schmidt Basics

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

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

- (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**

$$V(V^T V)^{-1} V^T \vec{w} = U(U^T U)^{-1} U^T \vec{w}. \quad (5)$$

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?*)

#### 4. Upper Triangularization

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

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \quad (6)$$

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

$$A = \begin{bmatrix} \vec{x} & \vec{y} & \vec{z} \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & a & b \\ 0 & \lambda_2 & c \\ 0 & 0 & \lambda_3 \end{bmatrix} \cdot \begin{bmatrix} \vec{x}^\top \\ \vec{y}^\top \\ \vec{z}^\top \end{bmatrix} \quad (7)$$

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.

$$\begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ \sqrt{2} \\ 0 \end{bmatrix}. \quad (8)$$

We also know that

$$\left\{ \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \quad (9)$$

is an orthonormal basis, and

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & -1 & 2 \\ 3 & -1 & 6 \\ -2 & 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -2\sqrt{2} \\ -2\sqrt{2} & -2 \end{bmatrix}. \quad (10)$$

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.

### 5. 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:

$$V = [\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n] \quad (11)$$

$$R = \begin{bmatrix} \vec{r}_1^\top \\ \vec{r}_2^\top \\ \vdots \\ \vec{r}_n^\top \end{bmatrix} \quad (12)$$

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

$$\vec{r}_1^\top = [\lambda_1 \quad r_{1,2} \quad r_{1,3} \quad \dots \quad r_{1,n}] \quad (13)$$

$$\vec{r}_2^\top = [0, \lambda_2, r_{2,3}, r_{2,4}, \dots \quad r_{2,n}] \quad (14)$$

$$\vec{r}_i^\top = \left[ \underbrace{0, \dots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \dots, r_{i,n} \right] \quad (15)$$

$$\vec{r}_n^\top = \left[ \underbrace{0, \dots, 0}_{n-1 \text{ times}}, \lambda_n \right] \quad (16)$$

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:

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

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

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 `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

$$\frac{d}{dt} y(t) = \lambda y(t) + \check{u}(t) \quad (19)$$

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**

$$\frac{d}{dt} \vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \quad (20)$$

$$\vec{x}(0) = \vec{x}_0, \quad (21)$$

for  $\vec{x}(t)$  by filling in the following template in the spots marked ♣, ◇, ♥, ♠.

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

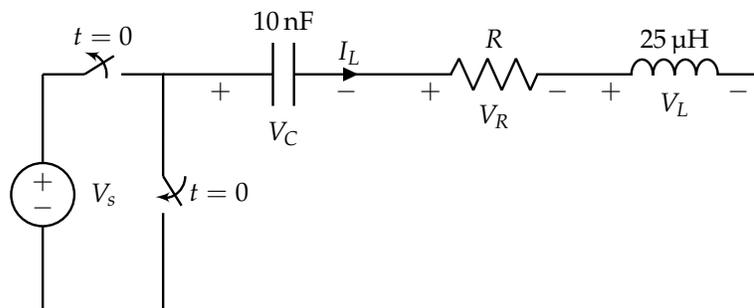
(HINT: The process here should be similar to diagonalization with some modifications. Start from the last row of the system and work your way up to understand the algorithm.)

- 1:  $\vec{\tilde{x}}_0 = V^\top \vec{x}_0$  ▷ Change the initial condition to be in  $V$ -coordinates
- 2:  $\vec{\tilde{u}} = V^\top \vec{u}$  ▷ 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:  $\check{u}_i = \clubsuit + \sum_{j=i+1}^n \spadesuit$  ▷ Make the effective input for this level
- 5:  $\tilde{x}_i = \text{ScalarSolve}(\diamond, \tilde{x}_{0,i}, \check{u}_i)$  ▷ Solve this level's scalar differential equation
- 6: **end for**
- 7:  $\vec{x}(t) = \heartsuit \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} (t)$  ▷ Change back into original coordinates

- (a) Give the expression for ♥ on line 7 of the algorithm above. (i.e., how do you get from  $\vec{\tilde{x}}(t)$  to  $\vec{x}(t)$ ?)
- (b) Give the expression for ◇ 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 ♣ on line 4 of the algorithm above.
- (d) Give the expression for ♠ on line 4 of the algorithm above.

## 6. 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  $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t)$  with a  $2 \times 2$  matrix  $A$ :

$$\begin{bmatrix} \frac{d}{dt}x_1(t) \\ \frac{d}{dt}x_2(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad (22)$$

where

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}. \quad (23)$$

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

$$\lambda_1 = -\frac{1}{2} \frac{R}{L} + \frac{1}{2} \sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \quad (24)$$

$$\lambda_2 = -\frac{1}{2} \frac{R}{L} - \frac{1}{2} \sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \quad (25)$$

- 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.
- 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  $V = [\vec{v}_{\lambda_1} \quad \vec{v}_{\lambda_2}]$  where  $\vec{v}_{\lambda_1}$  and  $\vec{v}_{\lambda_2}$  are the linearly independent eigenvectors of  $A$ . We could then diagonalize the matrix  $A$  as  $\tilde{A} = V^{-1}AV = \Lambda$  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  $V = [\vec{v}_{\lambda_1} \quad \vec{v}_{\lambda_2}]$  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  $U = [\vec{u}_1 \quad \vec{u}_2]$ . 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  $T = U^T A U$ .

- (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:

$$A = \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \quad (26)$$

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  $U$ :

$$U = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \quad (27)$$

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  $\tilde{x}$  in the transformed coordinates such that  $\vec{x}(t) = U\tilde{x}(t)$ . **Starting from  $\frac{d}{dt}\vec{x}(t) = A\vec{x}(t)$ , show that the differential equation in the transformed coordinate system is  $\frac{d}{dt}\tilde{x}(t) = T\tilde{x}(t)$**
- (e) Notice that the second differential equation for  $\frac{d}{dt}\tilde{x}_2(t)$  in the above coordinate system only depends on  $\tilde{x}_2(t)$  itself. There is no cross-term dependence. This happened because we transformed into a coordinate system which triangularizes  $A$ . **Compute the initial condition for  $\tilde{x}_2(0)$  and write out the solution to this scalar differential equation for  $\tilde{x}_2(t)$  for  $t \geq 0$ .** Assume that  $V_s = 1$  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  $\tilde{x}_2(t)$  in hand, use this to back-substitute and **write out the resulting scalar differential equation for  $\tilde{x}_1(t)$** . This should effectively have a time-dependent input (i.e.  $\tilde{x}_2(t)$ ) in it. **Also compute the initial condition for  $\tilde{x}_1(0)$**  (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  $\tilde{x}_1(t)$  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.

## 7. (PRACTICE) 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  $\{\vec{a}_1, \vec{a}_2, \dots, \vec{a}_n\}$  and run the following Gram-Schmidt algorithm on it to perform orthonormalization. It produces the vectors  $\{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n\}$ .

```

1: for  $i = 1$  up to  $n$  do ▷ Iterate through the vectors
2:    $\vec{r}_i = \vec{a}_i - \sum_{j < i} \vec{q}_j (\vec{q}_j^\top \vec{a}_i)$  ▷ Find the amount of  $\vec{a}_i$  that remains after we project
3:   if  $\vec{r}_i = \vec{0}$  then
4:      $\vec{q}_i = \vec{0}$ 
5:   else
6:      $\vec{q}_i = \frac{\vec{r}_i}{\|\vec{r}_i\|}$  ▷ Normalize the vector.
7:   end if
8: 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  $1 \leq \ell \leq n$ ,  $\text{Span}(\{\vec{a}_1, \dots, \vec{a}_\ell\}) = \text{Span}(\{\vec{q}_1, \dots, \vec{q}_\ell\})$ .
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  $\text{Span}(\{\vec{a}_1, \dots, \vec{a}_\ell\}) = \text{Span}(\{\vec{q}_1, \dots, \vec{q}_\ell\})$ . 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  $\text{Span}(\{\vec{a}_1\}) = \text{Span}(\{\vec{q}_1\})$ , then so too is  $\text{Span}(\{\vec{a}_1, \vec{a}_2\}) = \text{Span}(\{\vec{q}_1, \vec{q}_2\})$ , and so forth, until we get that  $\text{Span}(\{\vec{a}_1, \dots, \vec{a}_n\}) = \text{Span}(\{\vec{q}_1, \dots, \vec{q}_n\})$ .

(HINT: What you need to show is: if there exists  $\vec{\alpha} = [\alpha_1 \ \dots \ \alpha_k] \neq \vec{0}_k$  so that  $\vec{y} = \sum_{j=1}^k \alpha_j \vec{a}_j$ , then there exists  $\vec{\beta} = [\beta_1 \ \dots \ \beta_k] \neq \vec{0}_k$  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  $\{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n\}$ . 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  $1, 2, \dots, \ell$  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  $A = [\vec{a}_1 \ \cdots \ \vec{a}_n]$  and  $Q = [\vec{q}_1 \ \cdots \ \vec{q}_n]$ . If  $A$  and  $Q$  have the same column span then there exists square matrix  $U = [\vec{u}_1 \ \cdots \ \vec{u}_n]$  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  $\vec{a}_i = \sum_{j=1}^n \vec{u}_i \vec{q}_j$ . What does the algorithm tell you about this relationship? Can you figure out what  $\vec{u}_i$  should be?)

## 8. Homework Process, Study Group, and Course Weekly Survey

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.

At the same time, we want to check-in weekly regarding Discussions, Lectures, Lab, and Office Hours and see how effective they have all been for you as students.

**Please fill out this survey [link](#). For your submission, please attach a screenshot indicating that you have completed the survey this week.**

### Contributors:

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