EECS 16B Designing Information Systems and Devices II UC Berkeley Fall 2023

Homework 10

This homework is due on Saturday, November 4, 2023, at 11:59PM. Self-grades and HW Resubmissions are due on the following Saturday, November 11, 2023, at 11:59PM.

1. Correctness of the Gram-Schmidt Algorithm

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 $1 \leq \ell \leq n$, $\text{Span}(\{\vec{a}_1, \ldots, \vec{a}_\ell\}) = \text{Span}(\{\vec{q}_1, \ldots, \vec{q}_\ell\})$.
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 ℓ from 1 to n, and showing the statement that $\text{Span}(\{\vec{a}_1, \ldots, \vec{a}_\ell\}) = \text{Span}(\{\vec{q}_1, \ldots, \vec{q}_\ell\})$. 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 $\text{Span}(\{\vec{a}_1\}) = \text{Span}(\{\vec{q}_1\})$, 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 $\text{Span}(\{\vec{a}_1, \ldots, \vec{a}_n\}) = \text{Span}(\{\vec{q}_1, \ldots, \vec{q}_n\})$.

(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 ℓ 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 ℓ 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 ℓ 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 $\vec{q}_k$.)

2. Schur Decomposition Algorithm Application

Use the Schur Decomposition Algorithm to upper triangularize the following matrix:

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

You may use the fact that an eigenvector of A is $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, and that an eigenvector of $\begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. The algorithm is shown below for your reference:

Algorithm 1 Real Schur Decomposition

Require: A square matrix $A \in \mathbb{R}^{n\times n}$ with real eigenvalues.
Ensure: An orthonormal matrix $U \in \mathbb{R}^{n\times n}$ and an upper-triangular matrix $T \in \mathbb{R}^{n\times n}$ such that $A = UTU^\top$.

function REALSCHURDECOMPOSITION(A)
  if A is 1×1 **then**
    return $\begin{bmatrix} 1 \end{bmatrix}, A$
  end if
  $(\vec{q}_1, \lambda_1) :=$ FINDEIGENVECTOREIGENVALUE(A)
  $Q :=$ EXTENDBASIS($\{\vec{q}_1\}, \mathbb{R}^n$) ▷ Extend $\{\vec{q}_1\}$ to a basis of $\mathbb{R}^n$ using Gram-Schmidt; see Note 11
  Unpack $Q := \begin{bmatrix} \vec{q}_1 & \widetilde{Q} \end{bmatrix}$
  Compute and unpack $Q^\top A Q = \begin{bmatrix} \lambda_1 & \vec{\tilde{a}}_{12}^\top \\ \vec{0}_{n-1} & \widetilde{A}_{22} \end{bmatrix}$
  $(P, \widetilde{T}) :=$ REALSCHURDECOMPOSITION($\widetilde{A}_{22}$)
  $U := \begin{bmatrix} \vec{q}_1 & \widetilde{Q}P \end{bmatrix}$
  $T := \begin{bmatrix} \lambda_1 & \vec{\tilde{a}}_{12}^\top P \\ \vec{0}_{n-1} & \widetilde{T} \end{bmatrix}$
  return (U, T)
end function

You are welcome to use a calculator/computer for any matrix multiplication steps.

3. Using Upper-Triangularization to Solve Differential Equations

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

$$U = \begin{bmatrix}\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_n\end{bmatrix} \tag{2}$$

$$R = \begin{bmatrix}\vec{r}_1^\top \\ \vec{r}_2^\top \\ \vdots \\ \vec{r}_n^\top\end{bmatrix} \tag{3}$$

where the rows of the upper-triangular R look like

$$\vec{r}_1^\top = \begin{bmatrix}\lambda_1 & r_{1,2} & r_{1,3} & \dots & r_{1,n}\end{bmatrix} \tag{4}$$

$$\vec{r}_2^\top = \begin{bmatrix}0, \lambda_2, r_{2,3}, r_{2,4}, \dots & r_{2,n}\end{bmatrix} \tag{5}$$

$$\vec{r}_i^\top = \begin{bmatrix}\underbrace{0, \dots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \dots, r_{i,n}\end{bmatrix} \tag{6}$$

$$\vec{r}_n^\top = \begin{bmatrix}\underbrace{0, \dots, 0}_{n-1 \text{ times}}, \lambda_n\end{bmatrix} \tag{7}$$

where the λ_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:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{8}$$

$$\vec{x}(0) = \vec{x}_0, \tag{9}$$

where $\vec{x}_0$ is a specified initial condition and $\vec{u}(t)$ is a given vector of functions of time. (Note: $u(t)$ is not the same as the columns of U above)

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

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

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \lambda y(t) + \breve{u}(t) \tag{10}$$

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 U, R to construct a procedure for solving this differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{11}$$

$$\vec{x}(0) = \vec{x}_0, \tag{12}$$

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: $\vec{\widetilde{x}}_0 = U^\top \vec{x}_0$ ▷ Change the initial condition to be in V-coordinates
2: $\vec{\widetilde{u}} = U^\top \vec{u}$ ▷ 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 λ arguments to `ScalarSolve`, equation (2), for the i^{th} iteration of the for-loop?)

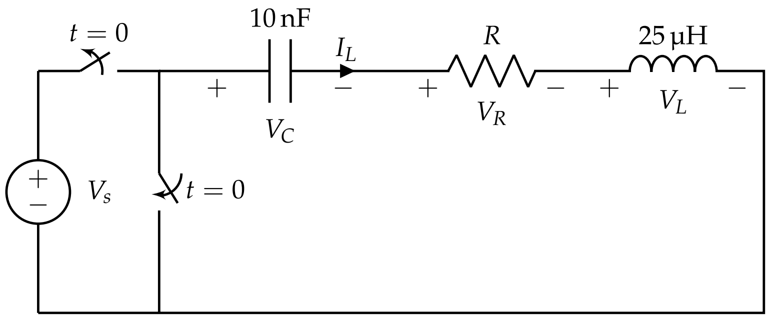
(HINT: Convert the differential equation to be in terms of R instead of A. It may be helpful to start with $i = n$ and develop a general form for the i^{th} row.)

(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.**

4. RLC Responses: Critically Damped Case

It is recommended that you complete the previous problem before starting this one. Consider the series RLC circuit below. 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. We can take the value of V_s as $V_s = 1\,\mathrm{V}$. For this problem, you may use a calculator/computer for calculations.

We can represent this circuit with the following vector differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}}_{A} \vec{x}(t) \tag{13}$$

where $\vec{x}(t) := \begin{bmatrix} I_L(t) \\ V_C(t) \end{bmatrix}$. We may calculate the eigenvalues of A symbolically as

$$\lambda_1 = -\frac{1}{2}\frac{R}{L} + \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{14}$$

$$\lambda_2 = -\frac{1}{2}\frac{R}{L} - \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{15}$$

(a) **Show that, if $R = 2\sqrt{\frac{L}{C}}$, then the two eigenvalues of A will be identical.**

(b) Using the previous part and the given values for capacitance and inductance, we find that our matrix is

$$A = \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \tag{16}$$

Show that the dimension of the eigenspace of $A - \lambda I$ is 1, where λ is the sole eigenvalue of A. Then, explain why we cannot use diagonalization. Here, $\lambda_1 = \lambda_2 = -2 \times 10^6$. Remember that we define the eigenspace of an eigenvalue to be $\mathrm{Null}(A - \lambda I)$.

(c) There are multiple ways to find an upper triangular matrix of A, and it is not unique. If you use the Schur decomposition method covered in lecture, you would find an upper triangular matrix R and the associated basis U for the system matrix A. For brevity, we will provide you with the basis U:

$$U = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{17}$$

Note that U is an orthonormal matrix. **Find the associated triangular matrix R.** 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 R we found. **Apply the algorithm you found in the previous problem to solve for $\vec{x}(t)$, given $I_L(0) = 0$ and $V_C(0) = V_S$.** Remember, $u(t) = 0$ in this case.

Contributors:

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