EECS 16B Designing Information Systems and Devices II UC Berkeley Fall 2022

Homework 10

This homework is due on Friday, November 4, 2022, at 11:59PM. Self-grades and HW Resubmissions are due on the following Friday, November 11, 2022, at 11:59PM.

1. Correctness of the Gram-Schmidt Algorithm

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: $\quad \vec{r}_i = \vec{a}_i - \sum_{j<i} \vec{q}_j \left(\vec{q}_j^\top \vec{a}_i\right)$ ▷ Find the amount of $\vec{a}_i$ that remains after we project
3: $\quad$ **if** $\vec{r}_i = \vec{0}$ **then**
4: $\qquad \vec{q}_i = \vec{0}$
5: $\quad$ **else**
6: $\qquad \vec{q}_i = \frac{\vec{r}_i}{\|\vec{r}_i\|}$ ▷ Normalize the vector.
7: $\quad$ **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 $\vec{q}_i \neq \vec{0}$.**

Solution: When $\vec{r}_i = \vec{0}$, then line 4 of the algorithm will make it terminate with $\vec{q}_i = \vec{0}$. This means that the nonzero case must come from the else statement and so from line 6 of the algorithm, we have $\vec{q}_i = \frac{\vec{r}_i}{\|\vec{r}_i\|}$. We know the norm of $\|\vec{q}_i\| = \vec{q}_i^\top \vec{q}_i$. Expanding this, we see

$$\|\vec{q}_i\|^2 = \vec{q}_i^\top \vec{q}_i = \left(\frac{\vec{r}_i}{\|\vec{r}_i\|}\right)^\top \left(\frac{\vec{r}_i}{\|\vec{r}_i\|}\right). \tag{1}$$

Grouping terms and using the definition of the norm, we get

$$\|\vec{q}_i\|^2 = \frac{1}{\|\vec{r}_i\|^2} \vec{r}_i^\top \vec{r}_i = \frac{1}{\|\vec{r}_i\|^2} \|\vec{r}_i\|^2 = 1. \tag{2}$$

(b) Next, we show the second property by considering each ℓ from 1 to n, and showing the statement that $\text{Span}(\{\vec{a}_1, \dots, \vec{a}_\ell\}) = \text{Span}(\{\vec{q}_1, \dots, \vec{q}_\ell\})$. 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 $\vec{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_k \end{bmatrix} \neq \vec{0}_k$ *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.)*

Solution: Proof outline: we first show that we can express $\vec{\beta}$ in terms of $\vec{\alpha}$, then vice versa. To do so, we express $\vec{a}_k$ or $\vec{q}_k$ using the algorithm above and then group $\vec{q}_j$ terms.

Assume for all $\ell \leq k-1$, for any $\alpha_1, \dots, \alpha_{k-1}$ there exists $\beta_1, \dots, \beta_{k-1}$ such that

$$\sum_{j=1}^{k-1} \beta_j \vec{q}_j = \sum_{j=1}^{k-1} \alpha_j \vec{a}_j \tag{3}$$

Then for $\ell = k$, consider a generic set of α's, $\alpha_1, \dots, \alpha_{k-1}, \alpha_k$.

$$\begin{aligned}
\sum_{j=1}^{k} \alpha_j \vec{a}_j &= \alpha_k \vec{a}_k + \sum_{j=1}^{k-1} \alpha_j \vec{a}_j && (4)\\
&= \alpha_k \left(\|\vec{r}_k\| \vec{q}_k + \sum_{i<k} \vec{q}_i \left(\vec{q}_i^{\top} \vec{a}_k \right) \right) + \sum_{j=1}^{k-1} \beta_j \vec{q}_j && (5)\\
&= \underbrace{\alpha_k \|\vec{r}_k\|}_{\beta_{k'}} \vec{q}_k + \sum_{j=1}^{k-1} \left(\underbrace{\beta_j + \alpha_k \left(\vec{q}_j^{\top} \vec{a}_k \right)}_{\beta_{j'}} \right) \vec{q}_j && (6)
\end{aligned}$$

The second equality is from line 2 of the algorithm where it states $\vec{r}_k = \vec{a}_k - \sum_{i<k} \vec{q}_i (\vec{q}_i^{\top} \vec{a}_k)$. This can be rearranged as

$$\vec{a}_k = \vec{r}_k + \sum_{i<k} \vec{q}_i \left(\vec{q}_i^{\top} \vec{a}_k \right) \tag{7}$$

This finishes the first direction of the proof, as we can now write the vector $\vec{y}$ as

$$\vec{y} = \sum_{j=1}^{k} \beta_{j'} \vec{q}_j. \tag{8}$$

Thus, it holds for $\ell = k$. The opposite direction follows similarly, which is shown below.

Once again, we assume that the statement holds for $\ell = k-1$. For any $\beta_1, \dots, \beta_{k-1}$ there exists $\alpha_1, \dots, \alpha_{k-1}$ such that

$$\sum_{j=1}^{k-1} \beta_j \vec{q}_j = \sum_{j=1}^{k-1} \alpha_j \vec{a}_j \tag{9}$$

Then for $\ell = k$, consider a generic set of β's, $\beta_1, \dots, \beta_{k-1}$.

$$\vec{y} = \sum_{j=1}^{k} \beta_j \vec{q}_j = \beta_k \vec{q}_k + \sum_{j=1}^{k-1} \beta_j \vec{q}_j \tag{10}$$

$$= \beta_k \frac{\vec{r}_k}{\|\vec{r}_k\|} + \sum_{j=1}^{k-1} \beta_j \vec{q}_j \tag{11}$$

$$= \frac{\beta_k}{\|\vec{r}_k\|}\left(\vec{a}_k - \sum_{j<k} \vec{q}_j \left(\vec{q}_j^\top \vec{a}_k\right)\right) + \sum_{j=1}^{k-1} \beta_j \vec{q}_j \tag{12}$$

$$= \frac{\beta_k}{\|\vec{r}_k\|}\vec{a}_k + \sum_{j=1}^{k-1} \vec{q}_j \left(\beta_j - \frac{\beta_k}{\|\vec{r}_k\|}\left(\vec{q}_j^\top \vec{a}_k\right)\right). \tag{13}$$

Notice that the second sum is only up through $k-1$ and involves a linear combination of the $\vec{q}_j$. So we can use our assumption: there exists an appropriate set of α that would weight $\vec{a}_j$ to equal that second sum.

$$\vec{y} = \frac{\beta_k}{\|\vec{r}_k\|}\vec{a}_k + \sum_{j=1}^{k-1} \alpha_j \vec{a}_j \tag{14}$$

Here, α_k can be defined to be $\frac{\beta_k}{\|\vec{r}_k\|}$ and we have proved the result for $\ell = k$. Thus, we have the new form of

$$\vec{y} = \sum_{j=1}^{k} \alpha_j \vec{a}_j. \tag{15}$$

(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 $\vec{q}_k$.)

Solution: The cases $i \leq k-1$ are given by the assumption, as stated in the hint. Then, for all $i \leq k-1$, for all $j < i$, $\vec{q}_j^\top \vec{q}_i = 0$. Remember that $\vec{q}_j^\top \vec{q}_j = 1$. All that remains is to deal with the case of $\vec{q}_k$ itself. We need to verify that it is orthogonal to all the previous $\vec{q}_j$. If $\|\vec{r}_k\| = 0$, then $\vec{q}_k = \vec{0}$ and so it definitely holds since the zero vector is orthogonal to every vector. So now we can assume $\|\vec{r}_k\| > 0$ and check for $i = k$.

For any $j < k$,

$$\vec{q}_j^\top \vec{q}_k = \vec{q}_j^\top \frac{1}{\|\vec{r}_k\|}\left(\vec{a}_k - \sum_{n<k} \vec{q}_n \left(\vec{q}_n^\top \vec{a}_k\right)\right) \tag{16}$$

$$= \frac{1}{\|\vec{r}_k\|}\left(\vec{q}_j^\top \vec{a}_k - \vec{q}_j^\top \sum_{n<k} \vec{q}_n \left(\vec{q}_n^\top \vec{a}_k\right)\right) \tag{17}$$

$$= \frac{1}{\|\vec{r}_k\|}\left(\vec{q}_j^\top \vec{a}_k - \sum_{n<k} \vec{q}_j^\top \vec{q}_n \left(\vec{q}_n^\top \vec{a}_k\right)\right) \tag{18}$$

$$= \frac{1}{\|\vec{r}_k\|}\left(\vec{q}_j^\top \vec{a}_k - \vec{q}_j^\top \vec{a}_k\right) = 0 \tag{19}$$

The final step is a cancellation of the cross terms, since they are all zero. Only when the index $n = j$ in the sum, the $\vec{q}_j^\top \vec{q}_j = 1$ and $\vec{q}_j^\top \vec{a}_k$ survives.

2. Schur Decomposition Algorithm Application

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

$$A = \begin{bmatrix} 1 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{20}$$

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 13
  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.

Solution: We are told that $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ is an eigenvector of A, so we can compute its eigenvalue as $A \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ so the eigenvalue is 1. We recognize that this is already a normalized vector, and we can extend the basis to be the standard basis. Hence, we obtain

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \tag{21}$$

Next, we unpack $Q^\top AQ = I^\top AI = A$. Using the notation from the notes, we have $\widetilde{\vec{a}}_{12}^\top = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$ and $\widetilde{A}_{22} = \begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$.

Next, we upper triangularize $\widetilde{A}_{22}$ (equivalently, we call REALSCHURDECOMPOSITION($\widetilde{A}_{22}$) at step 9). We are told that $\vec{1}1$ is an eigenvector of this matrix. We can normalize this to obtain $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$. We can extend the basis to get another orthonormal vector as $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$. Hence, $P = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$. We can unpack $P^\top \widetilde{A}_{22} P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. As we can see, we now have an upper triangular matrix, so we can return $P = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ and $\widetilde{T} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

Returning back to the problem of upper triangularizing A, we obtained $P = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ and $\widetilde{T} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ from calling REALSCHURDECOMPOSITION($\widetilde{A}_{22}$). Finishing up the algorithm on steps 10 and 11, we have

$$U = \begin{bmatrix} \vec{q}_1 & \widetilde{Q}P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{22}$$

and

$$T = \begin{bmatrix} \lambda_1 & \widetilde{\vec{a}}_{12}^\top P \\ \vec{0}_2 & \widetilde{T} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Note that the basis extension is non-unique, so you may end up with a different result. The important part is to make sure that you correctly perform the basis extensions and the steps of the Schur Decomposition Algorithm to obtain a valid decomposition.

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:

$$U = \begin{bmatrix}\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_n\end{bmatrix} \tag{24}$$

$$R = \begin{bmatrix}\vec{r}_1^\top \\ \vec{r}_2^\top \\ \vdots \\ \vec{r}_n^\top\end{bmatrix} \tag{25}$$

where the rows of the upper-triangular R look like

$$\vec{r}_1^\top = \begin{bmatrix}\lambda_1 & r_{1,2} & r_{1,3} & \dots & r_{1,n}\end{bmatrix} \tag{26}$$

$$\vec{r}_2^\top = \begin{bmatrix}0, \lambda_2, r_{2,3}, r_{2,4}, \dots & r_{2,n}\end{bmatrix} \tag{27}$$

$$\vec{r}_i^\top = \begin{bmatrix}\underbrace{0, \dots, 0}_{i-1 \text{ times}}, \lambda_i, r_{i,i+1}, r_{i,i+2}, \dots, r_{i,n}\end{bmatrix} \tag{28}$$

$$\vec{r}_n^\top = \begin{bmatrix}\underbrace{0, \dots, 0}_{n-1 \text{ times}}, \lambda_n\end{bmatrix} \tag{29}$$

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:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{30}$$

$$\vec{x}(0) = \vec{x}_0, \tag{31}$$

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 $\texttt{ScalarSolve}(\lambda, y_0, \check{u})$ that takes a real number λ, 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

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = \lambda y(t) + \check{u}(t) \tag{32}$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t), \tag{33}$$

$$\vec{x}(0) = \vec{x}_0, \tag{34}$$

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: $\widetilde{\vec{x}}_0 = U^\top \vec{x}_0$ ▷ Change the initial condition to be in V-coordinates
2: $\widetilde{\vec{u}} = U^\top \vec{u}$ ▷ 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: $\quad \breve{u}_i = \clubsuit + \sum_{j=i+1}^{n} \spadesuit$ ▷ Make the effective input for this level
5: $\quad \widetilde{x}_i =$ `ScalarSolve`$(\diamondsuit, \widetilde{x}_{0,i}, \breve{u}_i)$ ▷ 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 $\widetilde{\vec{x}}(t)$ to $\vec{x}(t)$?)

Solution: Since $\widetilde{\vec{x}}_0 = U^\top \vec{x}_0$ we know we are changing to U-basis. So, the implicit change of variable that we are doing is $\widetilde{\vec{x}} = U^\top \vec{x}$, this means that to come back, $\vec{x} = U\widetilde{\vec{x}}$ (since U is orthonormal, $(U^\top)^{-1} = U$). Thus, $\heartsuit = U$.

(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.)

Solution: We begin by taking our vector differential equation and substituting in our upper triangularization:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{u}(t) \tag{35}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = URU^\top\vec{x}(t) + \vec{u}(t) \tag{36}$$

Multiplying both sides by $U^\top$ and using the fact that $U^\top U = I$

$$U^\top \frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = RU^\top\vec{x}(t) + U^\top\vec{u}(t) \tag{37}$$

Now, we perform change of variables, $\widetilde{\vec{x}} = U^\top \vec{x}$ and $\widetilde{\vec{u}} = U^\top \vec{u}$ so we get,

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\vec{x}}(t) = R\widetilde{\vec{x}}(t) + \widetilde{\vec{u}}(t) \tag{38}$$

Thus, the i^{th} equation in this system is,

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{x}_i(t) = r_i^\top \widetilde{\vec{x}}(t) + \widetilde{u}_i(t) \tag{39}$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + r_{i,i+1}\widetilde{x}_{i+1}(t) + r_{i,i+2}\widetilde{x}_{i+2}(t) + \cdots + r_{i,n}\widetilde{x}_n(t) + \widetilde{u}_i(t) \tag{40}$$

$$= \lambda_i \widetilde{x}_i(t) + \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t) \tag{41}$$

Defining $\check{u}_i(t) = \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$ we can write eq. (41) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{x}_i(t) = \lambda_i \widetilde{x}_i(t) + \check{u}_i(t) \tag{42}$$

Here, we can see that when solving the scalar differential equation for the ith row, the scaling term is λ_i: $\diamondsuit = \lambda_i$.

(c) **Give the expression for ♣ on line 4 of the algorithm above.**

Solution: Since, from above, $\check{u}_i(t) = \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$ we can see that the $\widetilde{u}_i$ is the input term that does not depend on the inner sum. From this we conclude that $\clubsuit = \widetilde{u}_i$.

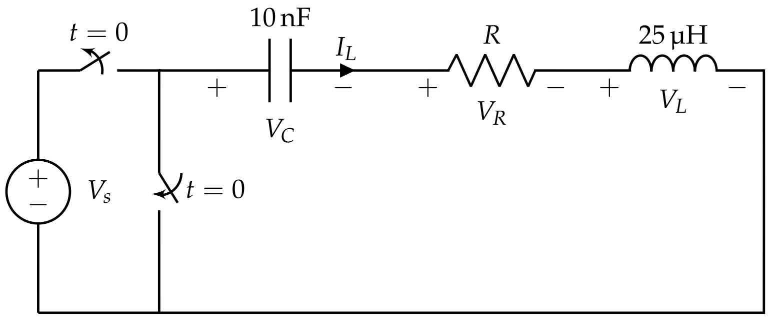
(d) **Give the expression for ♠ on line 4 of the algorithm above.**

Solution: Since, from above, $\check{u}_i(t) = \widetilde{u}_i(t) + \sum_{j=i+1}^{n} r_{i,j}\widetilde{x}_j(t)$ and so we know what is inside the inner sum: $\spadesuit = r_{i,j}\widetilde{x}_j$.

Congratulations! You now know how to systematically solve any system of differential equations with constant coefficients, as long as you know how to solve the scalar case with inputs. The same argument style applies for recurrence relations. The only gap that remains is the assumption that all the eigenvalues are real, but once you understand orthogonality for complex vectors, you can also update your understanding of upper-triangularization to allow for complex matrices as well.

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 $V_s = 1\,\text{V}$. For this problem, you may use a calculator/computer for calculations.

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}}_{A} \vec{x}(t) \tag{43}$$

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

$$\lambda_1 = -\frac{1}{2}\frac{R}{L} + \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{44}$$

$$\lambda_2 = -\frac{1}{2}\frac{R}{L} - \frac{1}{2}\sqrt{\left(\frac{R}{L}\right)^2 - \frac{4}{LC}} \tag{45}$$

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

Solution: If $R = 2\sqrt{\frac{L}{C}}$, the term in the square root will become

$$\frac{R^2}{L^2} - \frac{4}{LC} = 4\frac{L}{C}\cdot\frac{1}{L^2} - \frac{4}{LC} \tag{46}$$

$$= \frac{4}{LC} - \frac{4}{LC} = 0 \tag{47}$$

This means that

$$\lambda_1 = -\frac{1}{2L}\cdot 2\sqrt{\frac{L}{C}} = -\frac{1}{\sqrt{LC}} \tag{48}$$

$$\lambda_2 = -\frac{1}{2L}\cdot 2\sqrt{\frac{L}{C}} = -\frac{1}{\sqrt{LC}} \tag{49}$$

Hence, the eigenvalues are identitcal.

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

$$A = \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \tag{50}$$

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 $\text{Null}(A - \lambda I)$.

Solution: Our system's matrix becomes,

$$A = \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \tag{51}$$

Our eigenvalues λ_1 and λ_2 are identical, i.e.

$$\lambda_1 = \lambda_2 = \lambda = -\frac{R}{2L} = -2 \times 10^6 \tag{52}$$

Since the two eigenvalues are identical, we expect the corresponding eigenvectors $\vec{v}_1$ to be equal (or some scaled version of) $\vec{v}_2$. We can solve for $\vec{v}_1$ and $\vec{v}_2$ by finding the nullspace of $A - \lambda I$. Let's call the solution to this some generic eigenvector $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$.

$$\begin{bmatrix} -2 \times 10^6 & -4 \times 10^4 \\ 10^8 & 2 \times 10^6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{53}$$

Notice that the second column of the matrix above is $-50\times$ the first column. Therefore, $\vec{v}_1 = \vec{v}_2 = \begin{bmatrix} 1 \\ -50 \end{bmatrix}$. Furthermore, the nullspace is given by any multiple of the eigenvector, $\alpha \begin{bmatrix} 1 \\ -50 \end{bmatrix}$; we have a single dimensional nullspace and so this matrix A just has a one-dimensional eigenspace, even though the matrix A is 2×2.

We cannot use diagonalization since the dimension of our eigenspace is 1. When we are diagonalizing, we need our V matrix (with eigenvectors as columns) to be invertible. However, we can only find 1 linearly independent eigenvector so it cannot be invertible, as $V \in \mathbb{R}^{2\times 2}$.

(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:

$$U = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{54}$$

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.

Solution: The triangular matrix R is:

$$R = U^{-1}AU \tag{55}$$

Since U is an orthonormal matrix:

$$R = U^T AU \tag{56}$$

$$= \left(\frac{1}{\sqrt{2501}}\begin{bmatrix}1 & -50\\ 50 & 1\end{bmatrix}\right)\left(\begin{bmatrix}-4\times 10^6 & -4\times 10^4\\ 10^8 & 0\end{bmatrix}\right)\left(\frac{1}{\sqrt{2501}}\begin{bmatrix}1 & 50\\ -50 & 1\end{bmatrix}\right) \tag{57}$$

$$= \frac{1}{2501}\begin{bmatrix}1 & -50\\ 50 & 1\end{bmatrix}\begin{bmatrix}-4\times 10^6 & -4\times 10^4\\ 10^8 & 0\end{bmatrix}\begin{bmatrix}1 & 50\\ -50 & 1\end{bmatrix} \tag{58}$$

$$= \begin{bmatrix}-2\times 10^6 & -1.0004\times 10^8\\ 0 & -2\times 10^6\end{bmatrix} \tag{59}$$

It is ok if you rounded off -1.0004×10^8 to -1×10^8.

NOTE: The rest of the solution below is purely for informational purposes as a worked numerical example of the Schur Decomposition algorithm for this system. It is not asked for — and not required — in the student solution. However, the Schur Decomposition algorithm is within the scope of the course.

If we follow the Schur Decomposition algorithm, we can find the upper triangular matrix R and the associated basis U for the system matrix A. When extending our basis to $\mathbb{R}^2$ for the Gram-Schmidt procedure, we choose $\vec{e}_1 = \begin{bmatrix}1\\0\end{bmatrix}$ and $\vec{e}_2 = \begin{bmatrix}0\\1\end{bmatrix}$ (i.e. the columns of the 2×2 identity matrix).

For accuracy, we carry through any square roots until the end and be as precise with decimal points as possible.

We have shown already that our system matrix $A = \begin{bmatrix}-4\times 10^6 & -4\times 10^4\\ 10^8 & 0\end{bmatrix}$ has one independent eigenvector $v_\lambda = \begin{bmatrix}1\\-50\end{bmatrix}$ with eigenvalue $\lambda = -2 \times 10^6$. Following the Schur Decomposition algorithm, this means:

$$\vec{q}_1 = \begin{bmatrix}1\\-50\end{bmatrix} \tag{60}$$

$$\lambda_1 = -2\times 10^6 \tag{61}$$

Continuing the algorithm, we aim to extend $\vec{q}_1$ to a basis Q of $\mathbb{R}^2$ using the vectors $\vec{e}_1$ and $\vec{e}_2$ given in the problem. We must construct the matrix Q using the Gram-Schmidt procedure.

Gram-Schmidt($\vec{a}_1, \vec{a}_2, \vec{a}_3$) = Gram-Schmidt($\vec{q}_1, \vec{e}_1, \vec{e}_2$) = Gram-Schmidt($\begin{bmatrix}1\\-50\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$):

$$\vec{z}_1 = \vec{a}_1 = \begin{bmatrix}1\\-50\end{bmatrix} \tag{62}$$

$$\to \vec{p}_1 = \frac{\vec{z}_1}{||\vec{z}_1||} = \frac{1}{\sqrt{1^2+(-50)^2}}\begin{bmatrix}1\\-50\end{bmatrix} = \frac{1}{\sqrt{2501}}\begin{bmatrix}1\\-50\end{bmatrix} \tag{63}$$

$$\vec{z}_2 = \vec{a}_2 - \langle \vec{a}_2, \vec{p}_1\rangle \vec{p}_1 \tag{64}$$

$$= \begin{bmatrix}1\\0\end{bmatrix} - \left(\begin{bmatrix}1 & 0\end{bmatrix}\frac{1}{\sqrt{2501}}\begin{bmatrix}1\\-50\end{bmatrix}\right)\left(\frac{1}{\sqrt{2501}}\begin{bmatrix}1\\-50\end{bmatrix}\right) \tag{65}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{1}{2501}(1)\begin{bmatrix} 1 \\ -50 \end{bmatrix} \tag{66}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{2501} \\ \frac{-50}{2501} \end{bmatrix} \tag{67}$$

$$= \begin{bmatrix} \frac{2500}{2501} \\ \frac{50}{2501} \end{bmatrix} \tag{68}$$

$$\to \vec{p}_2 = \frac{\vec{z}_2}{||\vec{z}_2||} = \frac{1}{\sqrt{\left(\frac{2500}{2501}\right)^2 + \left(\frac{50}{2501}\right)^2}} \begin{bmatrix} \frac{2500}{2501} \\ \frac{50}{2501} \end{bmatrix} = \frac{2501}{50\sqrt{50^2+1^2}} \begin{bmatrix} \frac{2500}{2501} \\ \frac{50}{2501} \end{bmatrix} = \frac{1}{\sqrt{2501}} \begin{bmatrix} 50 \\ 1 \end{bmatrix} \tag{69}$$

$$\vec{z}_3 = \vec{a}_3 - (\langle \vec{a}_3, \vec{p}_2 \rangle \vec{p}_2 + \langle \vec{a}_3, \vec{p}_1 \rangle \vec{p}_1) \tag{70}$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \left[\left(\begin{bmatrix} 0 & 1 \end{bmatrix} \frac{1}{\sqrt{2501}} \begin{bmatrix} 50 \\ 1 \end{bmatrix} \right) \left(\frac{1}{\sqrt{2501}} \begin{bmatrix} 50 \\ 1 \end{bmatrix} \right) + \left(\begin{bmatrix} 0 & 1 \end{bmatrix} \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 \\ -50 \end{bmatrix} \right) \left(\frac{1}{\sqrt{2501}} \begin{bmatrix} 1 \\ -50 \end{bmatrix} \right) \right] \tag{71}$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \left(\left(\frac{1}{2501} \right) \begin{bmatrix} 50 \\ 1 \end{bmatrix} + \left(\frac{-50}{2501} \right) \begin{bmatrix} 1 \\ -50 \end{bmatrix} \right) \tag{72}$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} \frac{50}{2501} \\ \frac{1}{2501} \end{bmatrix} + \begin{bmatrix} \frac{-50}{2501} \\ \frac{2500}{2501} \end{bmatrix} \right) \tag{73}$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{74}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{75}$$

$$\to \vec{p}_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{76}$$

As expected, Gram-Schmit produces 2 non-zero vectors, $\vec{p}_1$ and $\vec{p}_2$, and a zero vector, $\vec{p}_3$. We discard the zero vectors and return:

$$\vec{q}_1 := \vec{p}_1 = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 \\ -50 \end{bmatrix} \tag{77}$$

$$\vec{q}_2 := \vec{p}_2 = \frac{1}{\sqrt{2501}} \begin{bmatrix} 50 \\ 1 \end{bmatrix} \tag{78}$$

We then return to our Schur Decompositon to construct our basis Q:

$$Q = \begin{bmatrix} \vec{q}_1 & \widetilde{Q} \end{bmatrix} = \begin{bmatrix} \vec{q}_1 & \vec{q}_2 \end{bmatrix} = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{79}$$

and can calculate $Q^T A Q$:

$$Q^T A Q = \left(\frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix}^T \right) \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \left(\frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \right) \tag{80}$$

$$= \frac{1}{2501} \begin{bmatrix} 1 & -50 \\ 50 & 1 \end{bmatrix} \begin{bmatrix} -4 \times 10^6 & -4 \times 10^4 \\ 10^8 & 0 \end{bmatrix} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{81}$$

$$= \begin{bmatrix} -2\times 10^6 & -1.0004\times 10^8 \\ 0 & -2\times 10^6 \end{bmatrix} \tag{82}$$

$$:= \begin{bmatrix} \lambda_1 & \vec{\tilde{a}}_{12}^T \\ \vec{0}_1 & \widetilde{A}_{22} \end{bmatrix} \tag{83}$$

We numerically solved the matrix multiplication using our favorite matrix calculator. First, we note that the top left entry is our $\lambda = -2\times 10^6$ of our system as was found previously. Next, we note $\vec{\tilde{a}}_{12}^T$ is our top right entry -1.0004×10^8 (try to keep the decimal points, but its fine if you didn't). Third, the bottom left entry is the (1-D) zero vector, as expected. Fourth, the bottom right entry is defined in the Schur Decomposition algorithm as the sub-matrix $\widetilde{A}_{22}$. For this case, this sub-matrix is simply a 1×1 matrix, i.e. $\widetilde{A}_{22} = \begin{bmatrix} -2\times 10^6 \end{bmatrix}$.

We now recursively call the Schur Decomposition algorithm on this sub-matrix $\widetilde{A}_{22}$. Since it is 1×1, we simply return:

$$P = 1 \tag{84}$$

$$\widetilde{T} = \widetilde{A}_{22} = \begin{bmatrix} -2\times 10^6 \end{bmatrix} \tag{85}$$

Returning from our recursive sub-call, we construct the change of basis matrix $U := \begin{bmatrix} \vec{q}_1 & \widetilde{Q}P \end{bmatrix}$. Recall that we had constructed $Q = \begin{bmatrix} \vec{q}_1 & \widetilde{Q} \end{bmatrix} = \begin{bmatrix} \vec{q}_1 & \vec{q}_2 \end{bmatrix}$, hence $\widetilde{Q} = \begin{bmatrix} \vec{q}_2 \end{bmatrix}$. Since $P = \begin{bmatrix} 1 \end{bmatrix}$, this means $\widetilde{Q}P = \vec{q}_2$ and hence:

$$U = \begin{bmatrix} \vec{q}_1 & \vec{q}_2 \end{bmatrix} = Q = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \tag{86}$$

Likewise, we can calculate our final triangular matrix T:

$$T := \begin{bmatrix} \lambda_1 & \vec{\tilde{a}}_{12}^T P \\ \vec{0}_1 & \widetilde{T} \end{bmatrix} = \begin{bmatrix} -2\times 10^6 & -1.0004\times 10^8 \\ 0 & -2\times 10^6 \end{bmatrix} \tag{87}$$

(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.

Solution: Recall that our upper triangular matrix R is

$$R = \begin{bmatrix} -2\times 10^6 & -1.0004\times 10^8 \\ 0 & -2\times 10^6 \end{bmatrix} \tag{88}$$

The initial condition for $\vec{\tilde{x}}(t) := \begin{bmatrix} \widetilde{I_L}(t) \\ \widetilde{V_C}(t) \end{bmatrix}$ is

$$\vec{\tilde{x}}(0) = U^\top \vec{x}(0) = \begin{bmatrix} \frac{-50}{\sqrt{2501}} \\ \frac{1}{\sqrt{2501}} \end{bmatrix} \tag{89}$$

First, we will solve for $\widetilde{V_C}(t)$. The differential equation and initial condition are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{V_C}(t) = -2\times 10^6 \widetilde{V_C}(t) \tag{90}$$

$$\widetilde{V_C}(0) = \frac{1}{\sqrt{2501}} \tag{91}$$

so

$$\widetilde{V_C}(t) = \frac{1}{\sqrt{2501}} e^{-2\times 10^6 t} \tag{92}$$

Next, we will solve for $\widetilde{I_L}(t)$. The differential equation and initial condition are given by

$$\frac{d}{dt}\widetilde{I_L}(t) = -2\times 10^6 \widetilde{I_L}(t) - 1.0004\times 10^8 \widetilde{V_C}(t) = -2\times 10^6 \widetilde{I_L}(t) - \frac{1.0004\times 10^8}{\sqrt{2501}} e^{-2\times 10^6 t} \tag{93}$$

$$\widetilde{I_L}(0) = \frac{-50}{\sqrt{2501}} \tag{94}$$

Hence,

$$\widetilde{I_L}(t) = \widetilde{I_L}(0) e^{-2\times 10^6 t} + \frac{(-1.0004\times 10^8) e^{-2\times 10^6 t}}{\sqrt{2501}} \int_0^t e^{(2\times 10^6 - 2\times 10^6)t'}\, dt' \tag{95}$$

$$= -\frac{50}{\sqrt{2501}} e^{-2\times 10^6 t} - \frac{1.0004\times 10^8}{\sqrt{2501}} t e^{-2\times 10^6 t} \tag{96}$$

This means that

$$\vec{\widetilde{x}}(t) = \begin{bmatrix} \widetilde{I_L}(t) \\ \widetilde{V_C}(t) \end{bmatrix} = \begin{bmatrix} -\frac{50}{\sqrt{2501}} e^{-2\times 10^6 t} - \frac{1.0004\times 10^8}{\sqrt{2501}} t e^{-2\times 10^6 t} \\ \frac{1}{\sqrt{2501}} e^{-2\times 10^6 t} \end{bmatrix} \tag{97}$$

Finally, we have $\vec{x}(t) = U\vec{\widetilde{x}}(t)$ so

$$\vec{x}(t) = \frac{1}{\sqrt{2501}} \begin{bmatrix} 1 & 50 \\ -50 & 1 \end{bmatrix} \begin{bmatrix} -\frac{50}{\sqrt{2501}} e^{-2\times 10^6 t} - \frac{1.0004\times 10^8}{\sqrt{2501}} t e^{-2\times 10^6 t} \\ \frac{1}{\sqrt{2501}} e^{-2\times 10^6 t} \end{bmatrix} \tag{98}$$

$$\approx \begin{bmatrix} -(4\times 10^4) t e^{-2\times 10^6 t} \\ (1 + 2\times 10^6 t) e^{-2\times 10^6 t} \end{bmatrix} \tag{99}$$

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.

