Problem 1: Coupled Ordinary Differential Equations

For the system,

\[ \begin{align*}
\frac{du_1}{dt} &= -u_1 + u_2 \\
\frac{du_2}{dt} &= -2u_1 - 4u_2
\end{align*} \]

a (10 points)
Use the method of substitution to derive a single second-order ODE for either \( u_1 \) or \( u_2 \) with respect to \( t \). Solve for the general solution \( u_1(t) \) and \( u_2(t) \).

b (10 points)
Now convert the system into matrix form,

\[ \frac{du}{dt} = A u \]

Define \( u \) and \( A \). Find the eigenvectors and eigenvalues of \( A \), and the general solution \( u(t) \).

c (5 points)
Explain how the two methods in part a and b are related.

d (10 points)
Decompose the matrix \( A \) you defined in part b into the product of three matrices, \( A = U \Lambda U^{-1} \), where \( \Lambda \) is a diagonal matrix. Find \( U \), \( \Lambda \), and \( U^{-1} \). How do these matrices relate to what you solved in part b? How is \( U^{-1} \) useful in finding the particular solution \( u(t) \) for given initial conditions \( u(0) \)?

e (5 points)
Plot the particular solution \( u_1(t) \) and \( u_2(t) \) from \( t = 0 \) to 10, for initial conditions \( u_1(0) = 0 \) and \( u_2(0) = 1 \).

f (10 points)
Now plot the Euler approximation of the ODE obtained by iterating

\[ u(t + \Delta t) = u(t) + \Delta t A u(t) \]

from the same initial conditions \( u_1(0) = 0 \) and \( u_2(0) = 1 \), for a time step size \( \Delta t = 0.1 \). What do you observe, and why?

Problem 2: Principal Component Analysis (PCA)

The goal of this problem is to provide you a tutorial of doing Principal Components Analysis (PCA) step by step at a very basic level, and then we’ll slightly increase the complexity of the problem and solve it computationally. This technique has become widely used in a variety fields, ranging from facial recognition to genetic analysis.

This problem might require looking up several statistical definitions, as they are critical in aiding your understanding of PCA. Furthermore, while all the plotting should be done via Matlab (or any other program), do the calculations by hand or by matrix algebra in Matlab, until you get the final parts of the problem. This means that you should not be using functions such as \( std, \ cov, \ eig, \ pca \), or \( scd \); only add, subtract,
multiply, transpose, etc. You will come to appreciate this situation where once you perform this step-by-step, you’ll better understand the manipulations you’re performing and why you’re performing them. Whether you choose to do this by hand or by simple data manipulation in Matlab, please turn in your work/code.

Lastly, as this problem is more of a tutorial, rather than of a straightforward question and answer format, we have written all the parts of the problem we want you to answer in boldface text.

a (5 points)

We will begin with two sets, each with \( p \) samples of data: \( x_1[j] \) and \( x_2[j] \), \( j = 1, \ldots, p \). The two sets pertain to two different conditions (you can imagine that these are the applications of two different antibiotics):

\[
x_1 = \begin{bmatrix} 5.9, 3.3, 5.1, 5.3, 6.5, 5.1, 4.8, 3.9, 4.4, 4.2 \end{bmatrix}
\]
\[
x_2 = \begin{bmatrix} 2.5, 0.4, 2.8, 2.2, 3.1, 2.7, 1.9, 1.0, 1.4, 0.9 \end{bmatrix}
\]

which can be conveniently grouped in a matrix \( X \) with rows \( x_1[j] \) and \( x_2[j] \).

To begin, calculate the standard deviation and variance of \( x_1 \) and \( x_2 \). Next, calculate the covariance of \( x_1 \) and \( x_2 \). The covariance formula is very similar to that for variance, except you’ll be using both \( x_1 \) and \( x_2 \), rather than just one of them. Either solve completely, or simply explain, what is the covariance of \( x_1 \) with itself? And is the covariance of \( x_1 \) with \( x_2 \) the same as the covariance of \( x_2 \) with \( x_1 \)?

Now that you’re able to calculate covariance, calculate the covariance matrix \( C \) of the above data, where each element \( C_{ij} \) is the covariance of \( x_i \) with \( x_j \). Hence, for an \( n \)-dimensional data set, the matrix \( C \) has \( n \) rows and \( n \) columns (\( n = 2 \) here). Note that down the main diagonal, the covariance is between a given dimension and itself.

b (10 points)

Once you have the covariance matrix, we begin to analyze the data in terms of its principal components. The first step is to calculate the eigenvalues \( \lambda_i \) and eigenvectors \( U_i \) of the covariance matrix, and be sure to normalize the eigenvectors as unit-norm eigenvectors, \( \|U_i\| = 1 \). The principal components \( T_i \) of the data \( X \) are defined as the projections of the data onto each of the eigenvectors \( U_i \).

Plot the data and the PCs (principal components), and see how they compare. How would you describe the PCs with respect to each other and the data? Which of the two PCs seems most important?

A common practice when performing PCA is dimensionality reduction: removing the PCs of lesser significance, i.e. those with substantially smaller variance, from their back-projection reconstruction of the data. This is a loss of information, but as these components do not explain much of the variance, their loss is often acceptable and helps simplify the analysis. Naturally, the first step then is to order the eigenvectors by eigenvalue, and then remove the components that you want to get rid of. This allows us to simplify the data set from \( n \) dimensions to much fewer dimensions.

Order the eigenvectors by eigenvalue, and then form the feature vector, which is just the matrix of vectors that you want to keep (hint: in this case, your feature vector should only be one vector, but generally your data will have higher dimensionality).

c (10 points)

Once you have your ordered eigenvectors and your chosen feature vector, you can create new data sets to represent your original data in terms of these principal components. You’ll probably notice that you cannot simply multiply your matrix of eigenvectors against your data, and will require to transpose at least some of the matrices to obtain the desired result. Create these new data sets for both the matrix of ordered eigenvectors and the matrix of feature eigenvectors. Describe the final result. Lastly, plot the transformed data involving the ordered eigenvector matrix. Plotting the other data would be unnecessary. Why?

---

1. Just as before, boldface symbols in equations denote vectors or matrices.
d (5 points)

The last part of this first analysis will be to get back data as if we started with the data expressed in terms of principal components. Using the transforms you just performed, modify the equations slightly and get the original data back. Naturally, you’ll get the original data back only if you use all of the ordered eigenvectors and the appropriate transformed data. However, do a similar process using the feature vector and the transformed data from the feature vector. Plot both of these "original" data sets. What is the difference between the two "original" data sets? What happened?

e (15 points)

You’ve now worked through a simple data set by hand, and you are ready to move onto a more highly dimensional data set. Here, you do not need to do anything by hand or step-by-step; use the functions provided by Matlab, Python, etc. The data is of females at least 21 years old and of Pima Indian heritage, looking at whether or not they have diabetes and factors that might be associated with the disease. The data set is available at the following link:


The goal here is to see which variables are most associated with diabetes. You will use a programming language of your choice to identify the highest values eigenvalues and eigenvectors and plot the data against the greatest principal components. In your write up, state the two greatest eigenvalues and which variables they belong to, and plot the data against the three greatest principal components. (hint: depending on the language you choose and programs you use, you might have to remove the index, or leftmost, column of data, as well as the variable names, or first row of data, to be able to easily implement code. Furthermore, refer to the eig and pca functions in Matlab, or if you choose to use Python, sklearn would be helpful).

f (5 points)

Lastly, think about the eigenvectors and eigenvalues in problems 1 and 2. Which ones are the most important? And how does their importance compare (i.e., what do the eigenvectors and eigenvalues in problem 1 tell you, versus what they tell you in problem 2)?